
بسم الله الرحمان الرحيم
تعلم HTML
من . . . . الاحتراف
السهولة
2010
بقلم
أيوب معتصم
WWW.DINARO.COM

محتويات الكتاب

الدرس الحادي عشر :النمادج[1]

الدرس الثاني عشر :النمادج[2]

الدرس الثالث عشر :الوسوم الخاصة ب MS Internet Explorer

كلمة المؤلف

ان التطور الشديد الذي شهدته لغة HTML ،والاعداد الهائلة التي تحاول ان تتعلم هذه اللغة الجميلة والسهلة كان لابد لي من انشاء كتاب يتحدث عن هذه اللغة.

وفي الحقيقة توجد كثير من الكتب تتحدث عن لغة HTML ،ولكن قليل منها يعلمك كيف تصبح مصمم صفحات ويب ناجح.

وفي هذا الكتاب سنسير خطوة خطوة يا عزيزي المتدرب.

ولقد اعتمدت على عدة مصادر وكتب ....وادعوا لهم بالخير والصحة والعطاء الدائم.

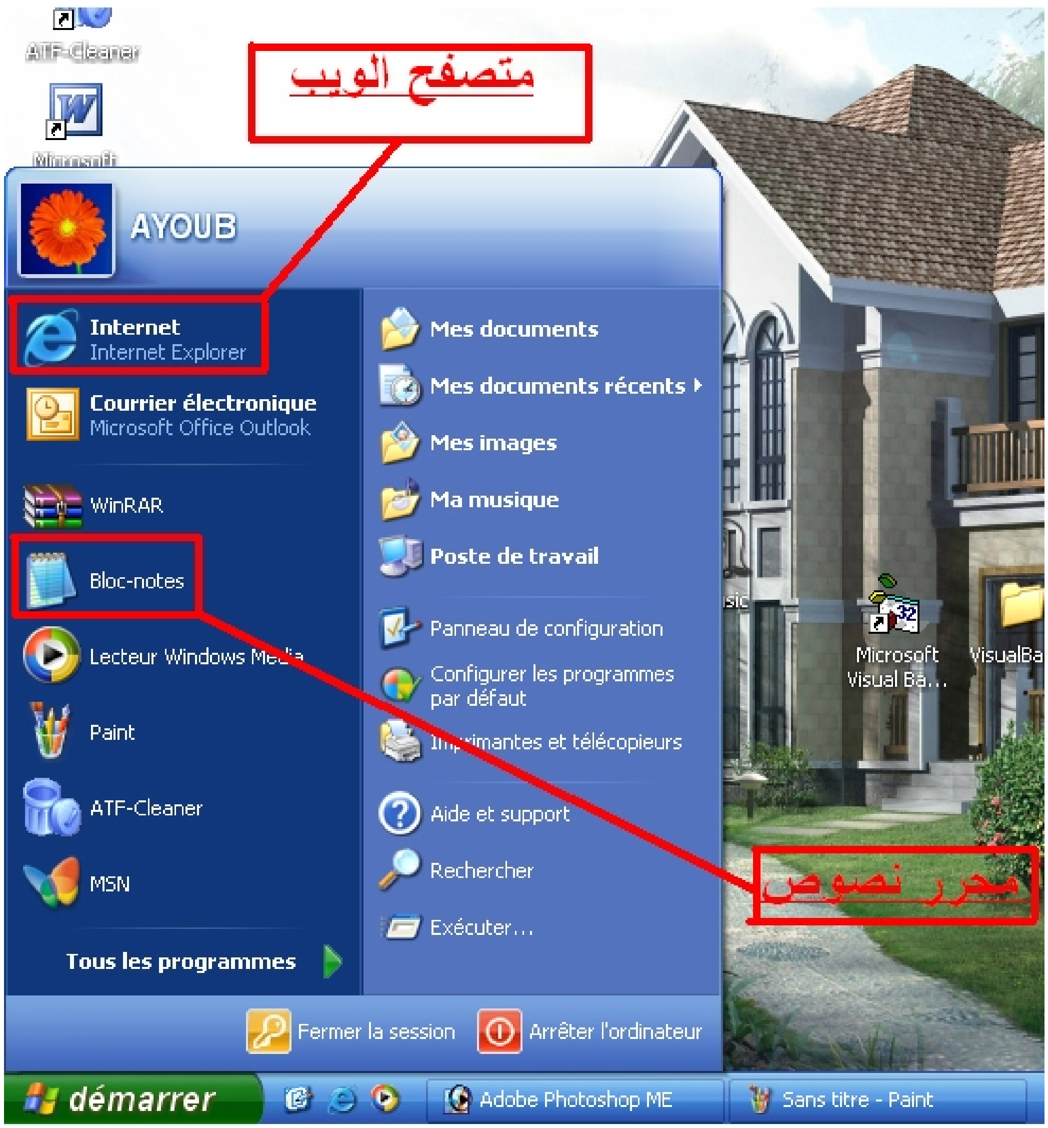
وعموما اتمنى لكم اسعد الاوقات، واستفادة موفقة بإذن الله !

المؤلف!!

أيوب معتصم

قبل قراءة هذا الكتاب

يجب ان يكون لديك محرر نصوص مثل NotePad المرفق مع Windows ومتصفح لمعاينة النتائج مثل Netscape Navigator أو MS Internet.



ولا تظن ان ابدا ان انشاء مواقع عملية صعبة هذا خاطئ هذا خاطئ هذا خاطئ......... واذا اكملت قراءة هذا الكتاب جيدا ستتمكن من تصميم صفحات الويب.

ان تعلم لغة HTML امر ممتع وستعرف ذلك.

يا عزيزي المتدرب لقد كنت مثلك مبتدئ لكن بعون الله ورعايته صرت مصمم مواقع ويب ناجح.

في حالة وجدت اي اخطاء في هذا الكتاب يمكنك مراسلتي على البريد التالي :

Dinaro2Ayoub@gmail.com

أو

Dinaro_22@hotmail.com

سيكون عما قريب بإذن الله كتب جديدة في عدة لغات منها : pascal و php و css و visual basic 6 و javascript و mysql.

ارجوا زيارة المنتدى *www.dinaro.com/vb3*

ثمن الكتاب

الدعاء لي ولوالدي ولجميع المسلمين بالصحة والهناء والازدهار والنماء والعطاء الدائم والشفاء..............................والفوز بالجنة والنجاة من النار وهذا كل ما أطلبه......وشكرا لكم...

بسم الله الرحمن الرحيم

سبحانك لا علم لنا إلا ما علمتنا إنك أنت العليم الحكيم

مقدمة في لغة HTML

أهلاً بك إلى مقدمة لغة HTML أو كما يسميها البعض لغة الهتمل .
حسناً فلنبدأ الشرح ولكن أولاً أريد أن أجيب على بعض أسئلتك واتوقع أن أول سؤال يقول :

ما معنى لغة HTML ؟

ان لغة HTML وهي اختصار لكلمة (Hyper Text Markup Language) وهي إحدى الطرق في تصميم صفحات على الانترنت أو تزيين الرسائل التي تبعث عبر البريدالالكتروني.

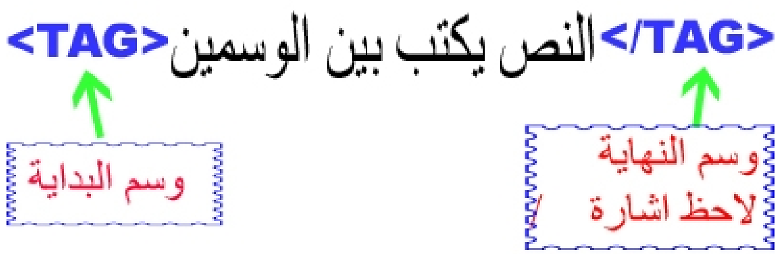
ان لغة HTML ليست لغة برمجية كلغة ++C أو JAVA أو C فهي لغة سهلة وغير معقدة ولا تحتوي على شيفرات كثيرة كذلك فهي لا تحتاج إلى مترجم . لذلك فهي لغة بسيطة جداً، وسهلة الفهم والتعلم ولا تحتاج لمعرفة مسبقة بلغات البرمجة والهيكلية المستخدمة فيها.

تحتاج دائماً إلى تطبيق ما تتعلمه بصورة عملية أكثر من مجرد الأمثلة المدرجة في الكتاب.

الدرس الاول

الأساسيات

حسناً فلنبدأ الشرح تكتب لغة HTML برموز غريبة وكثيرة اسمها باللغة الانجليزية الـTags وتسمى باللغة العربية الوسوم .



و لكل وسم من وسوم لغة الـHTML وسم بداية ووسم نـهاية وسوف امثله لك من خلال الرسم التالي :

وسم النهاية	وسم البداية
</html>	<html>
</HEAD>	<HEAD>
</TITLE>	<TITLE>
<BODY>	</BODY>

سوف أشرح كل وسم على حدى :

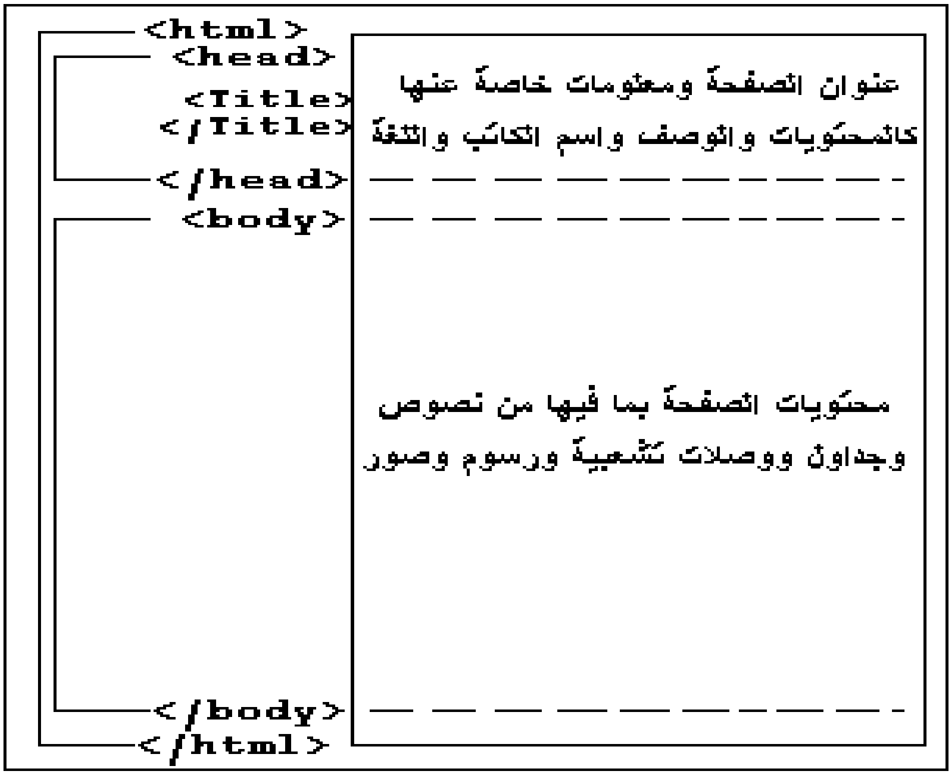
والوسم الأول هو الوسم < HTML > الذي كما ذكرت سابقاً أنه أهم الوسوم المستخدمة في بناء الصفحات المستخدم فيها لغة الـHTML ، لأنه يقوم بتعريف للمتصفح أن هذه الصفحة مصنوعة بلغة الـHTML ويوضع هذا الوسم في أول المستند المصنوع بلغة الـHTML ولولا هذا الوسم لظهرت محتويات الصفحة بصورة باهتة وغير مفهومة ، وسم النهاية هو < /HTML > الذي يوضع أخر المستند المصنوع بلغة الـHTML .

فلنأتي الآن للوسم < HEAD > الذي يستخدم لتعريف صاحب الصفحة واللغة والعديد من الأشياء التي سوف نشرحها في دروس قادمة إنشاء الله . وأذكرك أن وسم النهاية الخاص به هو < /HEAD >

نأتي الآن إلى الوسم < TITLE > الذي يمثل الجملة التي توضع أعلى الشاشة على اليسار ، ووسم النهاية الخاص به هو < /TITLE > ، ملاحظة يوضع هذا الوسم بين الوسمين < HEAD > و < /HEAD >

نأتي الآن إلى الوسم < BODY > الذي هو لب الدروس بأكملها ، حيث يمثل هذا الوسم محتويات الصفحة بأكملها حيث تضع فيه الجداول والصور و المواضيع وكل ماتريده أن يظهر على الصفحة .

سوف أبسط لك كل ما ذكرت في هذا الدرس داخل هذا الرسم التبياني :



إذن ملف Html يبدأ دائماً بالوسم <HTML> وينتهي بالوسم </HTML>. لا تنسى ذلك!

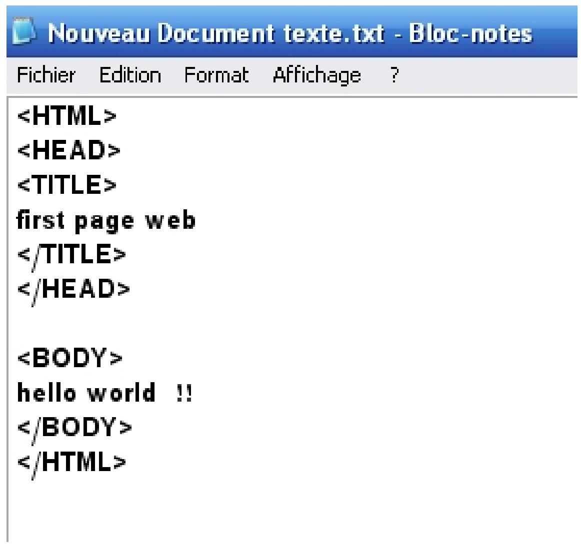
اذن عزيزي المتدرب ما رأيك لو نطبق ذلك بصورة عملية :

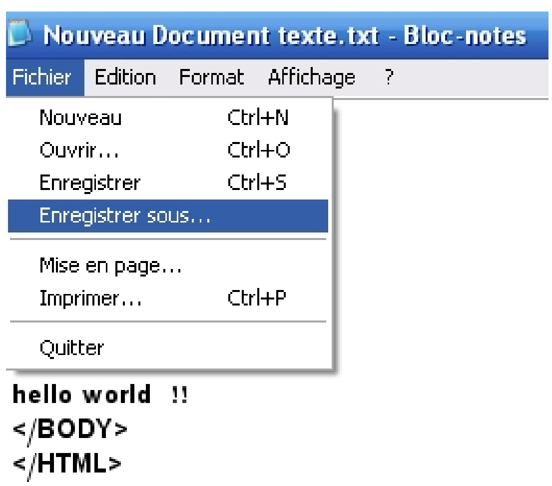
ولا تنسى أن الامتداد المستخدم في أسماء ملفات HTML هو *.htm* أو *.html*

مثال :

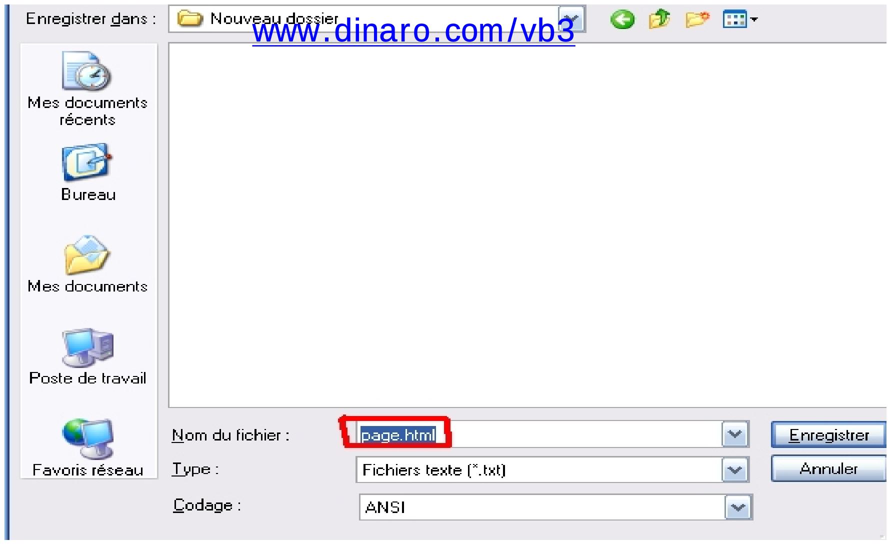
```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>
```

<BODY>
hello world !!
</BODY>
</HTML>

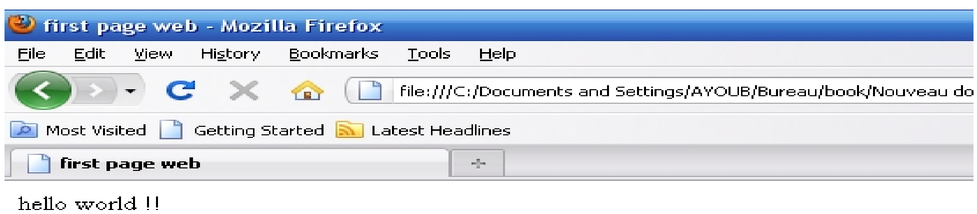




الناتج:



الناتج:



هل حصلت على نفس النتيجة؟ إذن مبروك لقد قمت بإنشاء أول صفحة ويب خاصة بك.

لا يوجد فرق بين كتابة الوسوم بالأحرف الإنجليزية الكبيرة UPPERCASE أو الأحرف الصغيرة lowercase. لذلك تستطيع الكتابة بأي شكل منهما أو حتى الكتابة بكليهما.

إن المتصفحات لا تأخذ بعين الاعتبار الفراغات الزائدة أو إشارات نهاية الفقرات (أي عندما تقوم بضغط مفتاح *Enter*) التي تجدها هذه المتصفحات في ملف Html.
وبعبارة أخرى فإن باستطاعتك كتابة ملفك السابق بالشكل التالي:

```
<HTML><HEAD><TITLE> first page web
</TITLE></HEAD><BODY>
hello world  !! </BODY></HTML>
```

أو بالشكل التالي:

```
<HTML>
<HEAD>
<TITLE>
first

page

web
</TITLE>
</HEAD>
<BODY>
hello

world
```

```
!!
</BODY>
</HTML>
```

وفي كل الحالات ستحصل على نفس النتيجة.

إذن كيف يمكن التحكم بمقدار النص المكتوب في كل سطر وكيف يمكن تحديد نهاية الفقرة وبداية الفقرة التي تليها؟

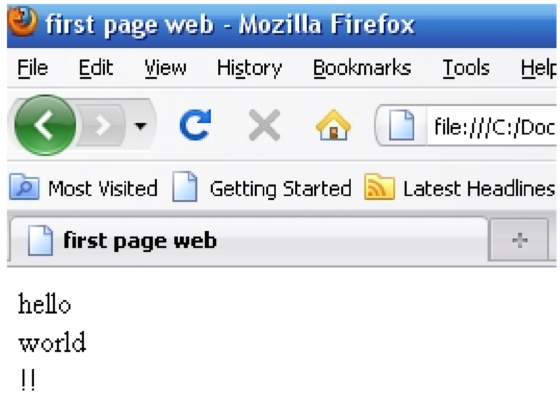
**سوف نستخدم الوسم
 لتحديد النهاية للسطر. والبدء بسطر جديد (لاحظ أن هذا الوسم مفرد، أي ليس له وسم نهاية).**

قم بتعديل الملف السابق لكي يصبح بالشكل التالي :

```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>

<BODY>
hello <BR> world <BR>  !!
</BODY>
</HTML>
```

الناتج:



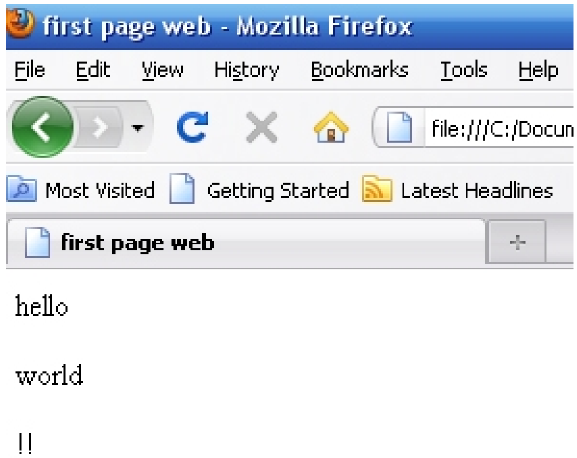
وهناك أيضا الوسم <**P**> الذي يقوم تقريباً بنفس عمل الوسم السابق أي أنه ينهي السطر أو الفقرة ويبدأ بسطر جديد لكن مع إضافة سطر إضافي فارغ بين الفقرات.

```
<HTML>

<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>

<BODY>
hello <p> world <p>  !!
</BODY>
</HTML>
```

الناتج:



أما الفراغات فتعتبر رموزاً خاصة لذلك لا نستطيع التحكم بها وبعددها إلا باستخدام الوسم   (والأحرف هي اختصار للعبارة Non Breakable Space). وإذا أردت إدخال عدة فراغات بين نص وآخر ما عليك إلا كتابة هذا الوسم بنفس عدد الفراغات المطلوب. كما يجب عليك التقيد بالأحرف الصغيرة هنا.

مثال :

```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>

<BODY>
hello     world      !!
</BODY>
</HTML>
```

الناتج:



هناك العديد من هذه الرموز والتي يجب أن تكتب بصورة معينة وباستخدام الوسوم وليس مباشرة بصورتـها العادية.

كل هذه الإشارات تستخدم أصلاً مع الوسوم فهي محجوزه ضمن مفردات لغة HTML ومن الخطأ استخدامها بصورتها الصريحة لئلا يؤدي ذلك إلى حدوث مشاكل في طريقة عرض الصفحة. كذلك فإن هناك رموزاً غير موجودة أساساً على لوحة المفاتيح كرمز حقوق الطبع © ورمز العلامة المسجلة ® ونحتاج إلى هذه الطريقة(طريقة الوسوم) لكتابتها.

وإليك جدول ببعض هذه الرموز ووسومها المكافئة. وألفت نظرك إلى أنها تكتب كما هي في الجدول وبدون إشارتي < >

الكـود	الــرمز
الــرمــوز	
‘	‘
’	’
‚	‚
“	“
”	”

×	×
÷	÷
–	–
—	—
¡	¡
¢	¢
£	£
¤	¤
¥	¥
€	€
¦	¦
§	§
▯	&#127;
¨	¨
©	©
ª	ª
«	«
¬	¬
®	®
¯	¯
°	°
±	±

ألیس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
H

الدرس الثاني

الألوان

أهلاً وسهلاً بك إلى الدرس الثاني من كتاب تعلم HTML.

سوف نقوم في هذا الدرس بالتعرف على الخصائص التي يمكن إضافتها إلى الوسم <BODY> من أجل التحكم بالشكل العام للصفحة، وخصوصا فيما يتعلق بالألوان.

طبعاً أنت لا زلت تذكر الصفحة التي قمنا بكتابتها في الدرس الأول. صفحة بسيطة بخلفية رمادية وخط صغير نسبياً لونه أسود. وهذه هي الإعدادات الإفتراضية التي يعتمدها المتصفح عندما لا نقوم نحن بتحديد إعدادات أخرى. (ربما تقول: أهذه صفحة إنترنت! أين الألوان والرسومات والخطوط الجميلة والتنسيقات التي نراها في صفحات الإنترنت؟ معك حق لكن مهلاً فما زلنا في البداية).

سوف نستمر باستخدام صفحتنا هذه لتوضيح أمثلة هذا الدرس أيضاً، لكن لن أقوم بتكرار كتابة وسوم البداية طالما أن عملنا يتركز في الجزء المخصص لمحتويات الصفحة نفسها أي ضمن الوسمين <BODY> ... </BODY>.

لنبدأ العمل يا عزيزي المتدرب !!!!!!

نطلق كلمة خاصية (Attribute) على التعابير التي تضاف إلى الوسوم، من أجل تحديد الكيفية أو الشكل الذي تعمل بها هذه الوسوم. وبعبارة أخرى فإن الوسم يقوم بإخبار المتصفح عن العمل الذي يجب القيام به أما الخاصية فتحدد الكيفية التي سيتم بها أداء هذا العمل.

مثال :

```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>

<BODY BGCOLOR="#00FF00">
hello world  !!
</BODY>
</HTML>
```

الناتج:



لقد قمت بإضافة الخاصية BGCOLOR إلى الوسم <BODY> ، وهي تقوم بتحديد لون الخلفية للصفحة. أما 00FF00 فهي القيمة التي تمثل اللون المختار وهو هنا اللون الأخضر، (لاحظ أنها مكتوبه بين إشارتي " ")

فمن أين جاءت هذه القيم، وكيف؟

<u>القليل عن الألوان...</u>

تلاحظ أن القيم السابقة مكونة من ستة رموز، وهي مكتوبة بالصيغة التالية:



هناك ثلاثة ألوان أساسية هي الأحمر والأخضر والأزرق، ولكل منها يوجد 256 درجة لونية ويعبر عن هذه الدرجات بالأرقام من 000 وحتى 255. ومن خلال مزج هذه الألوان بدرجاتها اللونية المختلفة نحصل على الألوان الأخرى.

فمثلا اللون الأسود مكون من الدرجة 000 من كل من اللون الأحمر والأخضر والأزرق. واللون الأبيض مكون من الدرجة 255 من هذه الألوان. أما اللون الأصفر فهو مكون من الدرجة 255 للون الأحمر، والدرجة 255 للون الأخضر، والدرجة 000 من اللون الأزرق... وهكذا بنفس الطريقة يتم تكوين باقي الألوان

وبعملية حسابية بسيطة 256×256×256 ينتج لدينا أن عدد الألوان التي يمكن الحصول عليها بمزج الألوان الثلاثة السابقة هو 16777216 بالضبط.

حسنا، لكن من أي جاءت الرموز FFFFFF والتي عبرت عن اللون الأبيض بها. إنها ببساطة أرقام... مكتوبة بالنظام السداس عشري (نظام عددي أساسه الرقم 16 ويعبر عنه باستخدام الأرقام العادية من 0 إلى 9 والرموز A,B,C,D,E,F). فالرقم 255 بالنظام العشري العادي يكافئه الرقم FF بالنظام السداس عشري.

إذن فالرقم السداس عشري FF على اليسار يمثل الدرجة 255 للون الأحمر. والرقم FF في الوسط يمثل الدرجة 255 من اللون الأخضر. والرقم FF على اليمين يمثل الدرجة 255 من اللون الأزرق.

وعلى هذا المنوال يعبر عن اللون الأزرق الفاتح بالرقم السداس عشري: CC6699 أما اللون الأسود فرقمه هو 000000.

وهذا جدول ببعض الألوان ورموزها المكافئة بالنظام السداس عشري.

Aqua #00FFFF	Antiquewhite #FAEBD7	Aliceblue #F0F8FF
Beige #F5F5DC	Azure #F0FFFF	Aquamarine #7FFFD4
Blanchedalmond #FFEBCD	Black #000000	Bisque #FFE4C4
Brown #A52A2A	Blueviolet #8A2BE2	Blue #0000FF
Chartreuse #7FFF00	Cadetblue #5F9EA0	Burlywood #DEB887
Cornflowerblue #6495ED	Coral #FF7F50	Chocolate #D2691E
Cyan #00FFFF	Crimson #DC143C	Cornsilk #FFF8DC
Darkgoldenrod #B8860B	Darkcyan #008B8B	Darkblue #00008B
Darkkhaki #BDB76B	Darkgreen #006400	Darkgray #A9A9A9
Darkorange #FF8C00	Darkolivegreen #556B2F	Darkmagenta #8B008B
Darksalmon #E9967A	Darkred #8B0000	Darkorchid #9932CC
Darkslategray #2F4F4F	Darkslateblue #483D8B	Darkseagreen #8FBC8F
Deeppink #FF1493	Darkviolet #9400D3	Darkturquoise #00CED1
Dodgerblue #1E90FF	Dimgray #696969	Deepskyblue #00BFFF
Forestgreen #228B22	Floralwhite #FFFAF0	Firebrick #B22222

Ghostwhite #F8F8FF	Gainsboro #DCDCDC	Fuchsia #FF00FF
Gray #808080	Goldenrod #DAA520	Gold #FFD700
Honeydew #F0FFF0	Greenyellow #ADFF2F	Green #008000
Indigo #4B0082	Indianred #CD5C5C	Hotpink #FF69B4
Lavender #E6E6FA	Khaki #F0E68C	Ivory #FFFFF0
Lemonchiffon #FFFACD	Lawngreen #7CFC00	Lavenderblush #FFF0F5
Lightcyan #E0FFFF	Lightcoral #F08080	Lightblue #ADD8E6
Lightgrey #D3D3D3	Lightgreen #90EE90	Lightgoldenrodyellow #FAFAD2
Lightseagreen #20B2AA	Lightsalmon #FFA07A	Lightpink #FFB6C1
Lightsteelblue #B0C4DE	Lightslategray #778899	Lightskyblue #87CEFA
Limegreen #32CD32	Lime #00FF00	Lightyellow #FFFFE0
Maroon #800000	Magenta #FF00FF	Linen #FAF0E6
Mediumorchid #BA55D3	Mediumblue #0000CD	Mediumauqamarine #66CDAA
Mediumslateblue #7B68EE	Mediumseagreen #3CB371	Mediumpurple #9370D8
Mediumvioletred #C71585	Mediumturquoise #48D1CC	Mediumspringgreen #00FA9A
Mistyrose #FFE4E1	Mintcream #F5FFFA	Midnightblue #191970
Navy #000080	Navajowhite #FFDEAD	Moccasin #FFE4B5

Orchid #DA70D6	Orangered #FF4500	Orange #FFA500
Paleturquoise #AFEEEE	Palegreen #98FB98	Palegoldenrod #EEE8AA
Peachpuff #FFDAB9	Papayawhip #FFEFD5	Palevioletred #D87093
Plum #DDA0DD	Pink #FFC0CB	Peru #CD853F
Red #FF0000	Purple #800080	Powderblue #B0E0E6
Saddlebrown #8B4513	Royalblue #4169E1	Rosybrown #BC8F8F
Seagreen #2E8B57	Sandybrown #F4A460	Salmon #FA8072
Silver #C0C0C0	Sienna #A0522D	Seashell #FFF5EE
Slategray #708090	Slateblue #6A5ACD	Skyblue #87CEEB
Steelblue #4682B4	Springgreen #00FF7F	Snow #FFFAFA
Thistle #D8BFD8	Teal #008080	Tan #D2B48C
Violet #EE82EE	Turquoise #40E0D0	Tomato #FF6347
Whitesmoke #F5F5F5	White #FFFFFF	Wheat #F5DEB3
Olive #808000	YellowGreen #9ACD32	Yellow #FFFF00

بيان بألوان الخلفيات :

#00FF00	#00CC00	#009900	#006600	#003300	#000000
#00FF33	#00CC33	#009933	#006633	#003333	#000033
#00FF66	#00CC66	#009966	#006666	#003366	#000066
#00FF99	#00CC99	#009999	#006699	#003399	#000099
#00FFCC	#00CCCC	#0099CC	#0066CC	#0033CC	#0000CC
#00FFFF	#00CCFF	#0099FF	#0066FF	#0033FF	#0000FF
#33FF00	#33CC00	#339900	#336600	#333300	#330000
#33FF33	#33CC33	#339933	#336633	#333333	#330033
#33FF66	#33CC66	#339966	#336666	#333366	#330066
#33FF99	#33CC99	#339999	#336699	#333399	#330099
#33FFCC	#33CCCC	#3399CC	#3366CC	#3333CC	#3300CC
#33FFFF	#33CCFF	#3399FF	#3366FF	#3333FF	#3300FF
#66FFCC	#66CC33	#669933	#666633	#663333	#660033
#66FF66	#66CC66	#669966	#666666	#663366	#660066
#66FF99	#66CC99	#669999	#666699	#663399	#660099
#66FFCC	#66CCCC	#6699CC	#6666CC	#6633CC	#6600CC
#66FFFF	#66CCFF	#6699FF	#6666FF	#6633FF	#6600FF
#99FF00	#99CC00	#999900	#996699	#993300	#990000
#99FF33	#99CC33	#999933	#996633	#993333	#990033
#99FF66	#99CC66	#999966	#996666	#993366	#990066
#99FF99	#99CC99	#999999	#996699	#993399	#990099
#99FFCC	#99CCCC	#9999CC	#9966CC	#9933CC	#9900CC
#99FFFF	#99CCFF	#9999FF	#9966FF	#9933FF	#9900FF
#CCFF00	#CCCC00	#CC9900	#CC6600	#CC3300	#CC0000
#CCFF33	#CCCC33	#CC9933	#CC6633	#CC3333	#CC0033
#CCFF66	#CCCC66	#CC9966	#CC6666	#CC3366	#CC0066
#CCFF99	#CCCC99	#CC9999	#CC6699	#CC3399	#CC0099
#CCFFCC	#CCCCCC	#CC99CC	#CC66CC	#CC33CC	#CC00CC
#CCFFFF	#CCCCFF	#CC99FF	#CC66FF	#CC33FF	#CC00FF
#FFFF00	#FFCC00	#FF9900	#FF6600	#FF3300	#FF0000
#FFFF33	#FFCC33	#FF9933	#FF6633	#FF3333	#FF0033
#FFFF66	#FFCC66	#FF9966	#FF6666	#FF3366	#FF0066
#FFFF99	#FFCC99	#FF9999	#FF6699	#FF3399	#FF0099
#FFFFCC	#FFCCCC	#FF99CC	#FF66CC	#FF33CC	#FF00CC
#FFFFFF	#FFCCFF	#FF99FF	#FF66FF	#FF33FF	#FF00FF

ملاحظة مهمة:

بعض المتصفحات لا تتعرف على رموز الألوان إلا بوضع إشارة # قبل هذه الرموز، لذلك من الأفضل استخدامها دائماً.

وبالنسبة لبعض الألوان الأساسية والدارجة، من الممكن استخدام أسماء هذه الألوان مباشرة بدلاً من الأرقام السداس عشرية. وهذا جدول يوضح هذه الألوان ومسمياتها:

White		Black	
Green		Red	
Purple		Marron	
Blue		Navy	
Lime		Teal	
Silver		Gray	
Aqua		Olive	
Yellow		Fuchsia	

ونعود إلى الوسوم و خصائصها ...

مثال :

```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>

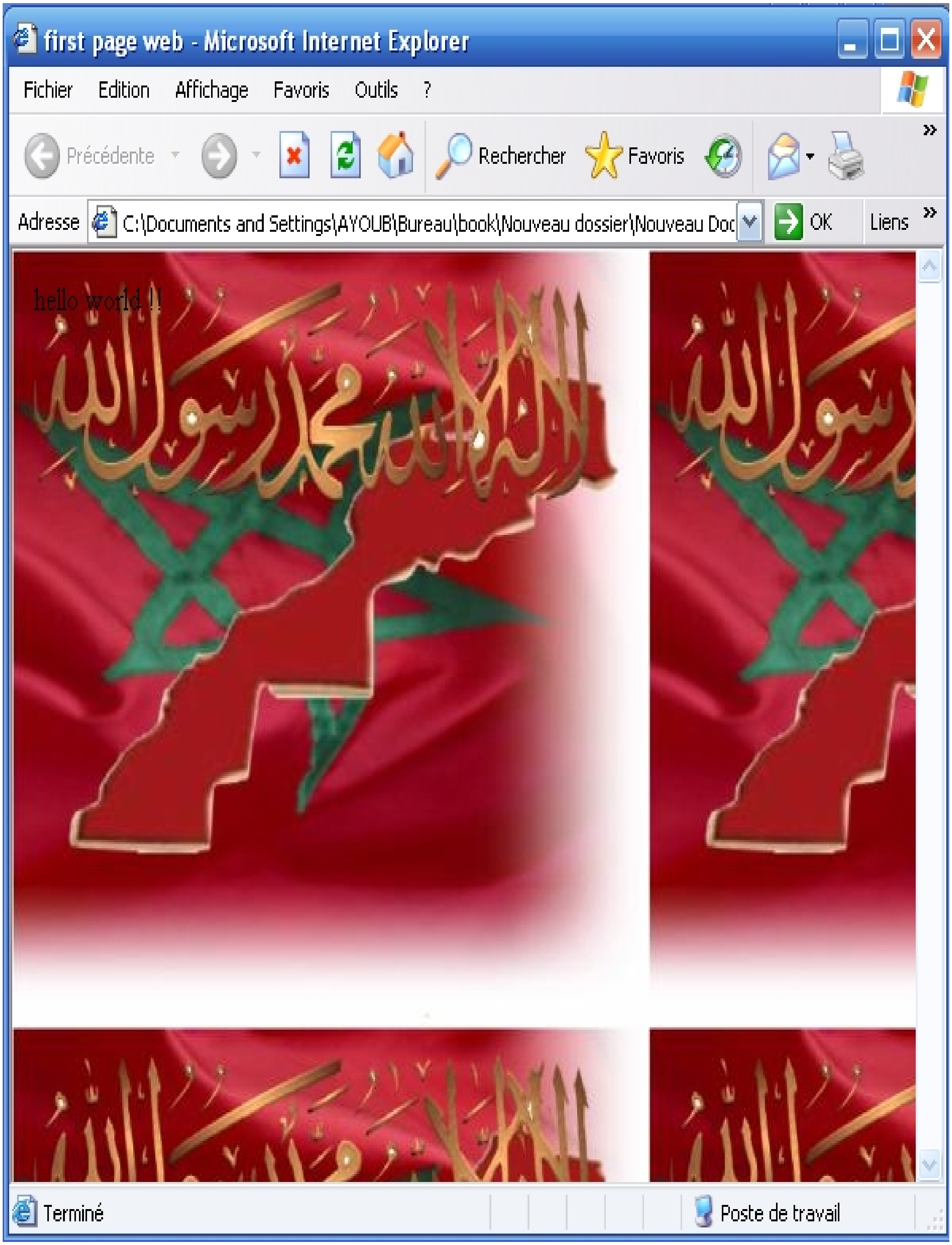
<BODY BGCOLOR="#000000"

 BACKGROUND=" maroc.jpg">
hello world  !!
</BODY>
</HTML>
```

تقوم الخاصية BACKGROUND بتحديد صورة كخلفية (ورق جدران) للصفحة وقد استخدمت الصورة التالية:



والمسماة maroc.jpg في صفحتي وكانت هذه النتيجة :



تلاحظ أن المتصفح قد قام بتكرار عرض الصورة بطريقة التجانب وأنها أصبحت تغطي كل الشاشة. بحيث حجبت أيضاً اللون الأبيض الذي حددناه كلون الخلفية (من خلال الخاصية BGCOLOR) والحقيقة أن اللون يظهر فقط عندما لا نقوم باستخدام صورة ما كخلفية. ومع ذلك يفضل تحديده إحتياطاً خاصة وأن بعض المتصفحات القديمة توصف بأنها متصفحات نصية Text-Based Browsers (أي ليس بإمكانها عرض الصور). أو ربما هناك بعض المستخدمين الذين قاموا بإلغاء خيار عرض الصور تلقائياً من متصفحاتهم. إذن لنعطهم على الأقل فرصة مشاهدة بعض الألوان إن لم يستطيعوا مشاهدة الصور.

إننا نستطيع استخدام الصور بأحجام مختلفة طولياً أو عرضياً كخلفيات للصفحة، والمتصفح نفسه هو الذي يقوم تلقائياً بعرضها في وضع التجانب مما يعطي الانطباع بأنها صورة كبيرة.

ولنكمل مع باقي الخصائص في وسم <BODY>: ربما لاحظت خلال استخدامك للإنترنت أن معظم الوصلات التشعبية (Links) التي تنقر عليها لتنقلك إلى صفحات أو مواقع أخرى على الشبكة هي دائماً مميزة باللون الأزرق، وأن الوصلات التي قمت بزيارتها فعلاً قد تحول لونها إلى القرمزي. حسناً، هذه هي الألوان الإفتراضية التي تعتمدها المتصفحات. لكن قد لا يعجبك ذلك وتريد تغيير هذا النظام. أو ببساطة ربما تريد استخدام لون أو صورة غامقة لخلفية الصفحة بما سيؤدي إلى اختفاء هذه الوصلات أو حتى اختفاء نص الصفحة نفسها. فما العمل؟

إليك هذه الخصائص التي تقوم بالتحكم في ألوان النصوص:

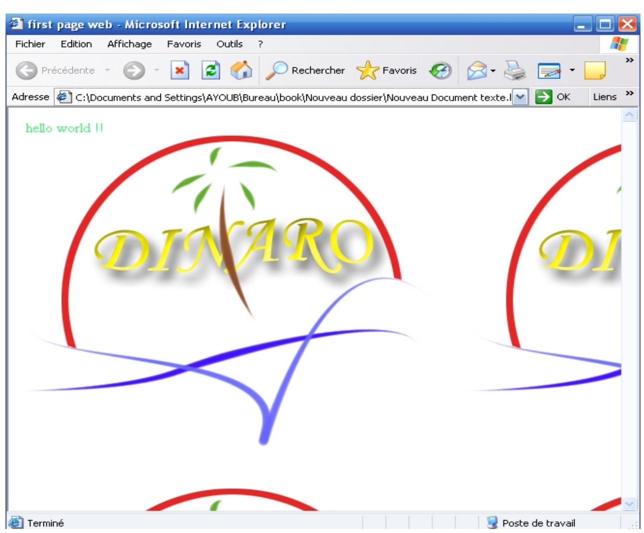
TEXT="#rrggbb"	تحديد لون النص الأساسي للصفحة
LINK="#rrggbb"	تحديد لون الوصلات التشعبية
VLINK="#rrggbb"	تحديد لون الوصلات التشعبية التي تمت زيارتها visited links
ALINK="#rrggbb"	تحديد لون الوصلة التشعبية الفعالة أي عندما يتم النقر عليها active links

والآن، دعنا نجمل الخصائص السابقة في عبارة واحدة. وسوف أكتب الرموز الخاصة بالألوان بنفس تلك الألوان التي تمثلها. وألفت نظرك إلى أنه لا أهمية للترتيب في كتابة هذه الخصائص داخل العبارة.

مثال :

```
<HTML>
<HEAD>
<TITLE>
first page web
</TITLE>
</HEAD>
<BODY BACKGROUND="dinaro.jpg"
BGCOLOR="#ff9900"
TEXT="#33ff66"
LINK="#6600ff"
VLINK="#ff0000"
ALINK="#999999">
hello world  !!
</BODY>
</HTML>
```

الناتج:



حاول أن تحللها! هل استنتجت أنني قد حددت الصورة dinaro.jpg كخلفية للصفحة؟ وأنني اخترت اللون البرتقالي للخلفية (في حالة عدم عرض الصورة السابقة كخلفية)؟ وان النص سيظهر باللون الأخضر ؟ أما الوصلات التشعبية فلونها أزرق، والوصلات التي تمت زيارتها ستظهر باللون الأحمر. أما تلك الوصلة الفعالة فستظهر باللون الرمادي في لحظة النقر عليها بالفأرة.

إذا كانت هذه هي استنتاجاتك... فمبروك، لقد نجحت. وكل ما أتمناه أن تكون قد قضيت وقتاً ملوناً وزاهياً مع هذا الدرس.

أليس هذا ممتعا

!!

HHHHHHHHHHHHHHHHHHHHHHHHH

الدرس الثالث

الخطوط

أهلاً وسهلاً بك إلى الدرس الثالث من كتاب تعلم HTML.

لا زلنا نناقش معاً أساسيات تنسيق صفحات الإنترنت والتحكم بخصائصها. وسوف نتابع ذلك في هذا الدرس من خلال التعرف على الوسوم الخاصة بالخطوط.

سوف تلاحظ في هذا الدرس والدروس اللاحقة أن هناك أكثر من طريقة لأداء نفس العمل، أو إعطاء نفس الخصائص لصفحات الإنترنت. وبالمقابل قد يبدو لك أن بعض الوسوم والخصائص متشابهة في تأثيرها، لكن بالقليل من التدقيق والتجربة ستكتشف أن لكل وسم خصوصيته.

لنبدأ العمل يا عزيزي المتدرب !!!!!!

راجع صفحتنا البسيطة التي عملنا فيها في الدرسين السابقين. إننا لم نقم بالتعامل مع الخطوط فيها ولا بأي شكل من الأشكال. أي أننا تركناها على إعداداتها الافتراضية.

وبالمناسبة فإن هذه الإعدادات هي خط عادي، نوعه Times New Roman وحجمه 3 (بمقياس متصفحات الإنترنت).

الوسم الأول الخاص بالخطوط هو <FONT> ... <FONT/>.

وهو يقوم بالتحكم بالخطوط من حيث النوع واللون والحجم. أما الخصائص التي نستخدمها مع هذا الوسم والوسوم الأخرى للخطوط فهي كالتالي:

Face	تقوم هذه الخاصية بتحديد نوع الخط الذي نريده، وقد نقوم بتحديد أكثر من نوع معاً. وفي هذه الحالة إذا لم يتواجد الخط المحدد أولاً على جهاز الشخص الذي يتصفح الموقع يتم إعتماد الخط الثاني ... وهكذا <FONT FACE="Traditional Arabic, Arabic Transparent, Simplified Arabic"> ... Text ... </FONT> طبعاً لا تنس أن تتأكد من كتابة أسماء الخطوط بالصورة الصحيحة هجائياً.
Color	أما هذه الخاصية فتحدد لون الخط، وذلك بنفس مبادئ تحديد الألوان التي تحدثنا عنها في الدرس السابق <FONT COLOR="#FF0000"> ... Text ... </FONT>
Size	ولتحديد حجم الخط نستخدم هذه الخاصية. وفقط هناك سبعة أحجام لأي خط تستطيع المتصفحات التعرف عليها. ونقوم بتحديد الحجم المطلوب بأسلوبين: أولهما المباشر. حيث يتم كتابة رقم يتراوح ما بين 1-7. أي أننا نختار الحجم الذي نريده مباشرة. <FONT SIZE="4"> ... Text ... </FONT>

وإليك نماذج بأحجام الخطوط

خط بحجم 1
خط بحجم 2
خط بحجم 3 (الخط الافتراضي)
خط بحجم 4
خط بحجم 5
خط بحجم 6
خط بحجم 7

أما الأسلوب الثاني فهو النسبي: حيث تكتب الأرقام من 1 إلى 6 مرفقة إما بإشارة + أو بإشارة −.

```
<FONT SIZE="+4">
... Text ...
</FONT>
```

وفي هذه الطريقة فإن الأرقام 1−6 تمثل درجات التكبير (+) أو التصغير (−) للخط وذلك نسبة إلى الحجم الافتراضي. فمثلا الرقم +4 يعني تكبير الخط أربع درجات عن الحجم الافتراضي وهو 3، أي أنه يصبح بالحجم 7. بالمقابل فأن الرقم −1 يعني تصغير الخط درجة واحدة أي يصبح بالحجم 2.
ولتوضيح هذا الأسلوب، إليك هذه النماذج:

خط بحجم −3
خط بحجم −2
خط بحجم −1
خط بحجم +0 (أو −0 وهو الافتراضي)
خط بحجم +1
خط بحجم +2
خط بحجم +3
خط بحجم +4
خط بحجم +5

لاحظ أنه حتى في الأسلوب النسبى لا نستطيع الحصول على أكثر من سبعة أحجام للخطوط.
حتى وإن حاولنا كتابة أرقام أكبر أو أصغر كما فعلت هنا بكتابة الحجم −3 أو +5.

والآن أعرف ماذا تريد أن تسأل، ستقول لقد ثبت حجم الخط على حده الأدنى عند الدرجة -2 وعلى حده الأعلى عند الدرجة +4. إذن ما الفائدة من وجود الدرجات الأخرى الأقل من -2 والأكبر من +4؟

حسنا وأنا أجيبك بسؤال آخر: ماذا لو قمنا بتغيير الحجم الافتراضي للخط في كل الصفحة إلى 1 بدلاً من 3؟ (وسوف نقوم بذلك فعلاً بعد قليل)، ألا نحتاج في هذه الحالة إلى الدرجات من +1 إلى +6 لتمثيل الأحجام الأكبر منه؟ وإذا قمنا بتحديد 7 كحجم إفتراضي ألا نحتاج إلى الدرجات من -1 إلى -6 لتمثيل الأحجام الأصغر منه؟ إذن نحن نحتاج فعلاً إلى هذه الدرجات لكي نغطي جميع الإحتمالات الواردة.

أرجو أن يكون هذا الجواب قد أقنعك :-)

وهذه بعض الأمثلة لتوضح لك كيفية استخدام هذا الوسم، وسوف أرفق نتيجة كل مثال بعده مباشرة.

<FONT FACE="arial" SIZE="6" COLOR="#6633ff”>

This font is Arial, Size is 6, Color is Red

</FONT>

This font is Arial, Size is 6, Color is Blue

<FONT FACE="arial" SIZE="+3" COLOR="#FF0000">
This font is Arial, Size is +3, Color is Red
</FONT>

This font is Arial, Size is +3, Color is Red

<FONT FACE="Arial" SIZE="5" COLOR="#00FF00"> This </FONT>
<FONT FACE="Times New Roman" SIZE="7" COLOR="#FF00FF"> is </FONT>
<FONT FACE="Arial" SIZE="2" COLOR="#FF0000"> multi </FONT>
<FONT FACE="Impact" SIZE="4" COLOR="#000000"> colors, </FONT>
<FONT FACE="Courier" SIZE="2" COLOR="#0000FF"> multi </FONT>
<FONT FACE="Times New Roman" SIZE="3" COLOR="#008080"> faces, </FONT>
<FONT FACE="Courier" SIZE="6" COLOR="#FFFF00"> and </FONT>
<FONT FACE="Arial" SIZE="5" COLOR="#808080"> multi </FONT>
<FONT FACE="Impact" SIZE="2" COLOR="#800000"> sizes </FONT>
<FONT FACE="Times New Roman" SIZE="7" COLOR="#00FFFF"> text </FONT>

This is multi colors, multi faces, and multi sizes text

<FONT FACE="Impact" SIZE="6" COLOR="#000000">C </FONT>
<FONT FACE="Impact" SIZE="6" COLOR="#008080">O</FONT>

```
<FONT FACE="Impact" SIZE="6"
COLOR="#FF0000">L</FONT>
<FONT FACE="Impact" SIZE="6"
COLOR="#0000FF">O</FONT>
<FONT FACE="Impact" SIZE="6"
COLOR="#800000">R</FONT>
<FONT FACE="Impact" SIZE="6"
COLOR="#FF00FF">S</FONT>
```

COLORS

ننتقل الآن إلى الوسم الثاني من الوسوم الخاصة بالخطوط وهو <BASEFONT>.

وعمله هو تحديد نوع الخط وخصائصه بالنسبة للصفحة كلها . أي أنه يقوم بتعريف نوع الخط الأساسي الذي سيستخدم في الصفحة من بدايتها إلى نهايتها ويحدد لونه وحجمه.

هل لاحظت انه وسم مفرد ولا يحتوي على وسم للنهاية؟ بالطبع ما الحاجة إلى وسم النهاية طالما أنه يتعامل مع الصفحة ككل ومع الإعدادات الأساسية لها، وليس مع كلمة أو سطر أو فقرة بذاتها. لذلك فإن هذا الوسم يكتب عادة في أول الملف، ويفضل مباشرة بعد وسم <BODY>. أما الخصائص المستخدمة معه فهي نفس الخصائص سالفة الذكر مع <FONT> ، (نستطيع استخدام الخاصية Name معه بدلاً من Face). وبنفس الطريقة وبدون أي اختلافات. وإليك هذه الشيفرة كمثال:

```
<BASEFONT Name="Arial" COLOR="#FF0000" SIZE="5">
```

وبدراسة هذا المثال نستنتج أنه يقوم بتعديل الخط الافتراضي للصفحة بحيث يصبح نوعه Arial وحجمه 5 ولونه أحمر. وبالتالي فإن كل النصوص المكتوبة في تلك الصفحة سيطبق عليها هذا النمط من الخط. ما لم نقم طبعا باستخدام الوسوم <Font> ... <Font/> لتعديلها والتحكم بمظهرها كما فعلنا في الأمثلة السابقة، فهي أكثر تحديداً وأكثر مرونة من الوسم <BASEFONT> .

وبمناسبة الحديث عن الألوان وتغيير اللون الأساسي لنص الصفحة. ألا تذكر أننا في الدرس السابق تكلمنا عن الخاصية Text التي تكتب مع الوسم <Body> والتي استخدمناها لتحديد لون نص الصفحة... أنا لا زلت أذكر ذلك.

لا يوجد تعارض بين هذه الخاصية وخاصية Color في الوسم <BASEFONT> فأنت بكل بساطة تستطيع استخدام أي منهما في صفحتك. وإذا حدث واستخدمت كلاهما فإن اللون المحدد مع الوسم <BASEFONT> هو الذي سيطبقه المتصفح ويعتمده.

وهنا أريد أن أذكرك بما قلته في بداية هذا الدرس:

يوجد دائماً أكثر من طريقة لأداء نفس العمل

هناك وسوم خاصة تستخدم لتمييز العناوين Headings في صفحات الإنترنت وهي:

</Hn> ... <Hn>

وحرف n هو رقم بين 1-6 يمثل مستوى العنوان.

<H1> Heading 1 </H1>
<H2> Heading 2 </H2>
<H3> Heading 3 </H3>
<H4> Heading 4 </H4>
<H5> Heading 5 </H5>
<H6> Heading 6 </H6>

Heading 1

Heading 2

Heading 3

Heading 4

Heading 5

Heading 6

ونأتي الآن إلى التنسيقات والتأثيرات التي يمكن إضافتها إلى النصوص. وفيما يلي الوسوم الخاصة بها متبوعة بمثال ونتيجته:

- **خط الغامق (الأسود العريض)، ونستخدم له الوسوم التالية:**

<B> ... </B>
<STRONG> ... </STRONG>

This is **Bold Text**	<B> Bold Text </B>
This is **Strong Text**	<STRONG> Strong Text </STRONG>

- **الخط المائل**

<I> ... </I>
<EM> ... </EM>

This is *Italic Text*	<I> Italic Text </I>
This is *Emphasized Text*	<EM> Emphasized Text </EM>

- **الخط المسطر**

<U> ... </U>

This is Undelined Text	<U> Undelined Text </U>

- **الخط المرتفع**

^{...}

This is $^{\text{Superscript Text}}$	^{Superscript Text}

ü الخط المنخفض

_{...}

This is Subscript Text	_{Subscript Text}

ü خط كبير

<BIG> ... </BIG>

This is Big Text	<BIG> Big Text </BIG>

ü خط صغير

<SMALL> ... </SMALL>

This is Small Text	<SMALL> Small Text </SMALL>

ü نص يعترضه خط

<STRIKE> ... </STRIKE>
<S> ... </S>

This is ~~Striked Text~~	<STRIKE> Striked Text </SRTIKE>
This is ~~Striked Text~~ too	<S> Striked Text </S>

ü نص الآلة الطابعة TeleType

<TT> ... </TT>

This is TeleType Text	<TT> TeleType Text </TT>

وهذا النص يعرف أيضاً بالنص موحَد المسافات Monospaced Text. ولتوضيح هذا المفهوم إليك المثال التالي:

إذا أخذنا الحرفين m,i وكتبنا كل منهما عشر مرات متتالية نلاحظ أن المساحة التي شغلها الحرف m هي أضعاف المساحة التي شغلها الحرف i

iiiiiiiiii
mmmmmmmmmm

أما عند استخدام الوسم <TT> ... </TT> فإن المساحة التي يشغلها كلا الحرفين تصبح موحدة

iiiiiiiiii
mmmmmmmmmm

وهذه أمثلة تجمع بين عدة تنسيقات معاً:

<B><I><U>
This is a Bold, Italic and Underlined Text
</U> </I> </B>

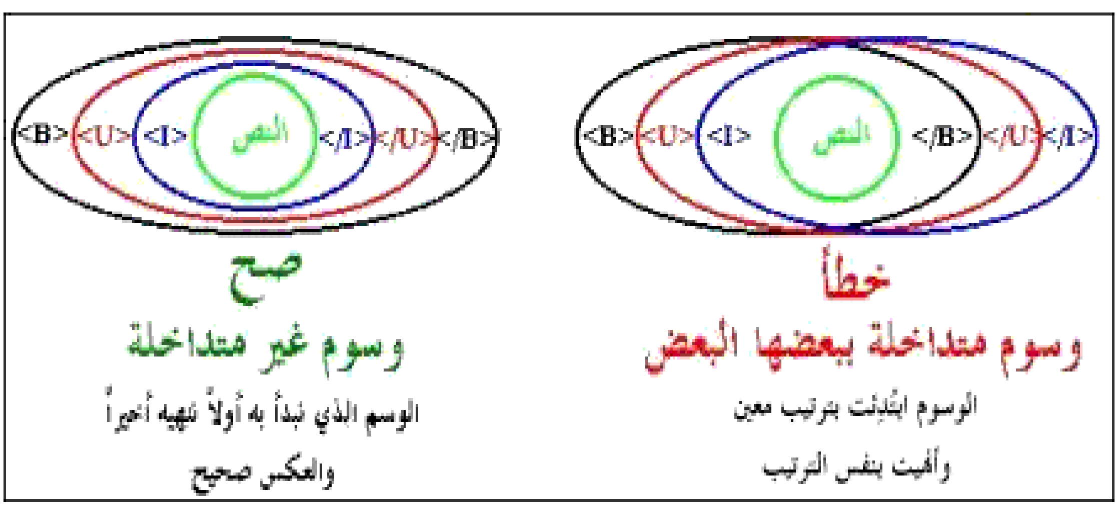
<u>This is a Bold, Italic and Underlined Text</u>

<FONT COLOR="#6633ff="+3"><U><I>
This text is red, size +3, Italic, and Underlined
</I> </U> </FONT>

This text is red, size +3, Italic, and Underlined

وقد أردت من هذه الأمثلة توضيح مسائل معينة أولها: أن بإمكاننا استخدام عدة وسوم وتنسيقات معاً في نفس الوقت ولنفس المقطع من النص. (وذلك لجميع الوسوم وليس فقط لوسوم الخطوط). وكما ذكرت سابقاً، لا أهمية لترتيب هذه الوسوم ولا أيها ورد أولاً... لكن

عند استخدام الوسوم المتعددة في مقطع واحد يجب مراعاة عدم التداخل بينها!... كيف؟ أنظر إلى الرسم التالي:



فكتابة الوسوم السابقة بالطرق التالية هو خطأ:

<B><I><U>
This is a Bold, Italic and Underlined Text
</B> </I> </U>

<B><I><U>
This is a Bold, Italic and Underlined Text
</B> </U> </I>

أعرف أنك لم تصدقني وأنك قمت بتجربة هذه الوسوم وربما حصلت على نتيجة صحيحة. حسناً العبرة ليست في عبارة واحدة مكونة من وسمين أو ثلاثة تكتبها في ملف صغير بل في صفحة إنترنت كاملة قد تتألف من مئات أو حتى آلاف الوسوم مكتوبة في ملف خالٍ من الأخطاء المنطقية والتداخلات التي قد تسبب الإرباك للمتصفحات، وتؤدي إلى عدم عرض هذه الصفحة بالشكل المناسب والمطلوب.

لذلك فأهمية أن تتجنب وجود الوسوم المتداخلة في صفحتك هو بنفس الأهمية التي يجب أن توليها لكتابة هذه الوسوم بالصورة الصحيحة إملائياً. وإلا فالمتصفحات لا ترحم. وكثيرة هي المرات التي حصل فيها المصممون على صفحات منهارة بسبب نسيان حرف واحد أو إشارة مثل < أو > أو "

بإختصار شديد... وكقاعدة أساسية، الصفحة المصممة جيداً هي الصفحة ذات الوسوم الصحيحة وغير المتداخلة.

وصلنا الآن إلى نهاية هذا الدرس. أتمنى أن تكون قد قضيت وقتاً ممتعاً معه. وأن لا يكون قد أحدث تداخلاً في وسوم أفكارك. أراك في الدرس التالي.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHH

الدرس الرابع

الفقرات والقوائم

أهلاً وسهلاً بك إلى الدرس الرابع من كتاب تعلم HTML.

في هذا الدرس سوف نناقش الوسوم الخاصة بالفقرات بشكل خاص وترتيب الصفحات وتنسيقها بشكل عام.

صحيح أن استخدامك للألوان والرسومات في الصفحة يضفي عليها نوعاً من الحيوية، وأن الخطوط تعطي صفحتك رونقاً وجمالاً. لكنك إن لم تهتم بترتيب صفحتك أو تقضي بعض الوقت في تنسيق هيكلها العام وتنظيم فقراتها وقوائمها، فإنه من الصعب عليك الحصول على صفحة ويب ناجحة. فالترتيب هو الخطوة الأولى لجذب اهتمام الزائر أو القارئ لصفحتك وتسهل عليه فهم الخطوط العريضة للصفحة.

**لقد قمت في الدرس الأول بإيضاح بعض الوسوم الخاصة بالفقرات. ولا بأس من تذكيرك بها. فالوسم <P> يقوم بإنهاء الفقرة. والوسم
 ينهي السطر الحالي وينقل النص إلى سطر جديد. والوسم   يقوم بإضافة الفراغات، ويجب تكرار كتابته بنفس عدد الفراغات المطلوب.**

ونتابع في هذا الدرس مع هذه الوسوم وغيرها.

لقد قلت إن الوسم <P> هو وسم مفرد لكنه يستخدم أيضاً كوسم مزدوج <P> ... </P> وفي هذه الحالة يمكّننا من تحديد إتجاه الفقرة وإتجاه النص فيها حيث يستخدم معه الخصائص ALIGN، DIR.

فالخاصية ALIGN تحدد محاذاة الفقرة وهي تأخذ القيم Left، Center, Right وأوضحها بالأمثلة التالية:

<P Align="left"> This is a left-aligned paragraph </P>

This is left-aligned paragraph

<P Align="right"> This is right-aligned paragraph</P>

This is a right-aligned paragraph

<P Align="center"> This is a centered paragraph</P>

This is a centered paragraph

كذلك لتوسيط الفقرات أو الكائنات بشكل عام في الصفحة نستطيع استخدام الوسوم <CENTER> ... <CENTER/>

<CENTER> This is a centered text </CENTER>

This is a centered text

أما الخاصية DIR والتي نستخدمها أيضاً مع <P> فتقوم بتحديد إتجاه قراءة النص وتأخذ القيم

LTR	إتجاه النص من اليسار إلى اليمين (Left To Right)
RTL	إتجاه النص من اليمين إلى اليسار (Right To Left)

(تذكر هذه الخاصية جيداً فهي مهمة عند كتابة صفحات باللغة العربية)

ولتنسيق الفقرات أيضاً يوجد الوسوم

<BLOCKQUOTE> ... </BLOCKQUOTE> أي وسوم الفقرات المقتبسة. ووظيفتها تمييز الفقرة من خلال إدراج مسافة إضافية على الهامشين الأيمن والأيسر لها.

أنظر إلى الفقرة التالية التي قمت (باقتباسها) من إحدى صفحات هذا الموقع. ومن ثم وضعتها ضمن <BLOCKQUOTE> ... </BLOCKQUOTE>

والحقيقة أنك تستطيع وضع عدة وسوم معاً إذا أردت إدراج هوامش أكبر. كما في المثال التالي:

<BLOCKQUOTE>
<BLOCKQUOTE>

النص يكتب هنا

</BLOCKQUOTE>
</BLOCKQUOTE>

وبالطبع ليس شرطاً أن تستخدم هذا الوسوم مع الفقرات المقتبسة فقط. فأنا مثلاً أضعها في بداية ونـهاية كل صفحة من صفحات هذا الموقع. وبالتالي يظهر النص بعيداً قليلاً عن حاشية الصفحة فهذا أفضل من أن يكون ملاصقاً لـها وأجمل.

والآن تأمل هذا الشكل وحاول أن تستنتج كيف قمت بإعداده...!؟

D	**C**	**B**	**A**
H	**G**	**F**	**E**
L	**K**	**J**	**I**
P	**O**	**N**	**M**
T	**S**	**R**	**Q**

**ربما توصلت إلى أني استخدمت عدداً كبيراً من وسوم الفراغات   ونـهاية السطر
.**

حسناً، إستنتاجك لا بأس به ولكنه ليس دقيقاً فأنا لم أستخدم أياً من هذه الوسوم هنا.

بل كل ما فعلته بعد إعداد هذا الشكل هو وضعه ضمن:

<PRE> ... </PRE>

وهما اختصار لكلمة Preformated أي المنسق مسبقاً. وبالفعل فقد احتفظ هذا الشكل بالتنسيق المسبق الذي تم إعداده به. لكن تم تحويل الخط إلى خط موحد المسافات (راجع الدرس السابق) ولو لم أقم بوضعه ضمن هذه الوسوم لكانت النتيجة كالتالي:

A B C D E F G H I J K L M N O P Q R S T

لاحظ أن هذا الوسم يستخدم مع الفقرات التي لا نحتاج فيها إلى تنسيقات متعددة للخطوط أو الألوان. بل فقط مع الفقرات العادية موحدة الخط والتنسيقات.

القوائم

تحتوي لغة HTML على مجموعة من الوسوم الخاصة بتنظيم البيانات في قوائم وباستخدام عدة خيارات. وهناك نوعين من القوائم:

أولهما المتسلسلة Lists Ordered.

واليك المثال التالي عليها

أسماء بعض المدن الفلسطينية :

1. ***القدس***
2. ***نابلس***
3. ***رام الله***
4. ***الخليل***
5. ***جنين***
6. ***طولكرم***

وثانيهما القوائم غير المتسلسلة Unordered Lists وهذا مثال عليها

أسماء بعض الجامعات الفلسطينية :

- ***جامعة النجاح***
- ***جامعة القدس المفتوحة***
- ***جامعة بيرزيت***
- ***جامعة الخليل***

عند التعامل مع القوائم بنوعيهما نحتاج إلى وسوم خاصة بتحديد بداية ونهاية القائمة ووسوم تحدد بنود هذه القائمة.

بالنسبة للقوائم المتسلسلة نستخدم الوسوم

</OL> ... <OL>

أما بالنسبة للقوائم غير المتسلسلة فنستخدم

</UL> ... <UL>

ولتعيين كل بند من بنود القائمة نستخدم الوسم <LI> وهو وسم مفرد يكتب في بداية السطر الخاص بكل بند List Item.

إذن عندما قمت بإنشاء القوائم السابقة استخدمت الشيفرة التالية:

```
<OL>
<LI>القدس
<LI>نابلس
<LI>رام الله
<LI>الخليل
<LI>جنين
<LI>طولكرم
</OL>
```

```
<UL>
<LI>جامعة النجاح
<LI>جامعة القدس المفتوحة
<LI>جامعة بيرزيت
<LI>جامعة الخليل
</UL>
```

والخاصية الوحيدة التي تستخدم مع هذه الوسوم هي TYPE ووظيفتها تحديد شكل الرمز الظاهر مع بنود القائمة، وعادة تستخدم مع وسوم بداية القوائم <UL> أو <OL> وبذلك نحدد رمزاً واحداً لكل القائمة.

ولكن نستطيع استخدامها أيضاً مع وسم البنود <LI> لإعطاء تحكم أكبر في مظهر القائمة من خلال تحديد رمز مختلف لكل بند.

فعند وضعها ضمن تعريف القوائم المتسلسلة تأخذ القيم: A, a, I, i التي تغير رموز الترقيم من الأرقام العادية الإفتراضية (والتي رمزها 1) إلى ترقيم باستخدام الأحرف اللاتينية الكبيرة أو الصغيرة، أو باستخدام الأرقام الرومانية كما ترى في الجدول التالي:

<OL TYPE="i">	<OL TYPE="I">	<OL TYPE="a">	<OL TYPE="A">
.i .ii .iii .iv .v	.I .II .III .IV .V	.a .b .c .d .e	.A .B .C .D .E

والحديث عن هذه الخاصية يقودني إلى الحديث عن مسألة مهمة في لغة HTML وهي مسألة الوسوم والخصائص المحددة بمتصفح معين دون غيره أي التي تعمل مع أحد المتصفحات ولا تعمل مع غيره.

والسبب في ذلك أن هذه الخاصية تستخدم أيضاً مع القوائم غير المتسلسلة، لكن ليس بصورة مطلقة..كيف؟ أنت ترى أن الرمز الموجود عند كل بند في القائمة هو عبارة عن نقطة سوداء يطلق عليها اسم Disc وهي المعرفة ضمناً في خاصية TYPE.
لكن هناك رموز أخرى يمكن إظهارها وهي المربع square، والدائرة المفرغة circle وتعرف بالشكل التالي:

<UL TYPE="square">
<UL TYPE="circle">

ولكن للأسف هذه الخاصية لا تعمل ولا يظهر تأثيرها إلا مع متصفح نيتسكيب وليس مع مايكروسوفت إكسبلورر الذي يتعامل فقط مع القيمة الإفتراضية للخاصية. (رجاءً لا يغضب مستخدمو إكسبلورر فهناك الكثير من الوسوم والخصائص التي لا يستطيع نيتسكيب عرضها أيضاً).

ولإتمام الحديث عن القوائم، أذكر لك أن هناك وسوماً أخرى تستخدم لإنشاء القوائم غير المتسلسلة، وبنفس الطريقة المستخدمة مع <UL>...</UL> وهذه الوسوم هي:

<DIR> ... </DIR>
<MENU> ... </MENU>

هناك نوع خاص من القوائم يدعى قوائم الشرح أو التعريفات Definition Lists وكما يدل الإسم تستخدم عندما نريد إدراج قائمة من المصطلحات يتبع كل واحد منها شرح أو تعليق.

HTML
Hyper Text Markup Language
WWW
World Wide Web
FTP
File Transfer Protocol
GIF
Graphical Interchange Format
JPG, JPEG
Joint Photographic Experts Group

ونحتاج لإنشاء هذه القوائم إلى ثلاثة وسوم:

الأول <DL> ... </DL> لتعريف بداية ونهاية القائمة.

والثاني <DT> ويوضع قبل كل مصطلح لتحديده، وهو وسم مفرد.

أما الثالث فهو <DD> وهو وسم الشرح أو التعليق وهو أيضا مفرد.

ولنقم الآن بكتابة شيفرة القائمة السابقة

```
<DL>
<DT>HTML <DD>Hyper Text Markup Language
<DT>WWW <DD>World Wide Web
<DT>FTP <DD>File Transport Protocol
<DT>GIF <DD>Graphical Interchange Format
```

<DT>JPG, JPEG <DD>Joint Photographic Experts Group
</DL>

وصلنا الآن إلى نـهاية هذا الدرس، والذي نـاقشنا فيه ترتيب الصفحات والفقرات والقوائم. أتمنى لك صفحات مرتبة دائماً.

أليس هذا ممتعا !!

HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH

الدرس الخامس

الصور والرسومات

أهلاً وسهلاً بك إلى الدرس الخامس من كتاب تعلم HTML.

في هذا الدرس سوف أقوم بالحديث عن الصور والرسومات وما يتعلق بالتعامل معها، بالإضافة إلى التعريف بأنواع الملفات الرسومية الدارجة في الإنترنت.

لقد اقتصر حديثنا عن الصور حتى الآن على إضافة خلفيات للصفحات، وكان ذلك في الدرس الثاني أما إدراج الصور ضمن الصفحات نفسها فله حكاية أخرى، أبدأ بروايتها لك الآن.

إن الوسم الرئيسي المستخدم لتعريف صورة ما داخل الصفحة هو <IMG> وهو وسم مفرد. لكن هل يكفي هذا لإدراج صورة؟ كلا، بالطبع يجب أن نحدد الصورة التي نريدها. لذلك نضيف الخاصية له SRC لتحديد موقع واسم الصورة.

الصورة التالية إسمها alah.jpg وعندما قمت بإدراجها. كانت الشيفرة الخاصة بذلك هي

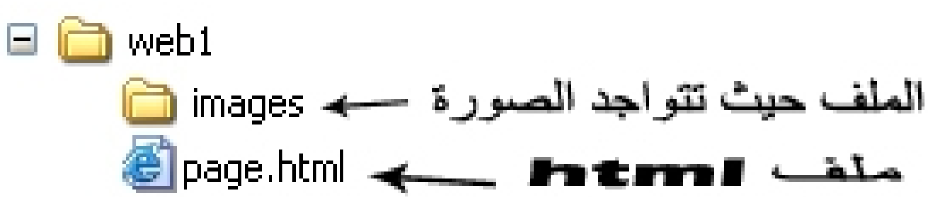
:

```
<IMG SRC=" alah.jpg ">
```



والصيغة هذه تفترض أن الصورة موجودة في نفس الدليل الفرعي أو المجلد حيث يتواجد ملف HTML الذي أعمل عليه، وقمت باستدعاء الصورة من خلاله. لكن ماذا لو كانت الصورة في مجلد فرعي آخر؟ حسنا سوف اناقش معك حالتين لهذه المسألة.

الحالة الأولى: أن تكون الصورة موجودة في مجلد متفرع عن المجلد الموجود به ملف HTML حسب الشكل التالي:



سطر الشيفرة

```
<IMG SRC="images/alah.jpg">
```

نقوم في هذه الحالة بكتابة إسم هذا المجلد تتبعه إشارة / ثم اسم الصورة.

الحالة الثانية: أن يكون ملف HTML موجوداً في مجلد ما وتكون الصورة موجودة في مجلد آخر بنفس المستوى. أي أنـهما مجلدين متجاورين وليسا متفرعين أحدهما عن الآخر.



وفي هذه الحالة نكتب .. (نقطتين) لتوجيه المتصفح للخروج من المجلد الفرعي الحالي (حيث يوجد ملف HTML) ومن ثم الدخول إلى المجلد images حيث توجد الصورة.

وبشكل عام، مهما كانت مواقع تواجد الملفات فإن عملية تحديد مواقعها والوصول إليها لا تخرج عن نطاق هذا النمط من الشيفرة. أي كتابة النقطتين للخروج من مجلد فرعي، وكتابة اسم المجلد الذي يجب الدخول إليه.

إن الأبعاد الأساسية لهذه الصورة هي 200×145 بيكسل Pixel (تابع القراءة حتى نـهاية هذا الدرس وأعدك أن أوضح لك ما هي وحدة البيكسل إذا كانت هذه أول مرة

تتعرف فيها على هذه الوحدة) وكما تلاحظ تم إدراج الصورة مع المحافظة على هذه الأبعاد. ومع ذلك فنحن نستطيع التحكم أيضاً بها وإظهار الصورة بالحجم الذي نريده من خلال هذا الوسم. كيف؟ بإضافة الخصائص HEIGHT, WIDTH متبوعة بأرقام تمثل الإرتفاع والعرض المطلوبين.

```
<IMG SRC="alah.jpg" HEIGHT="70" WIDTH="120">
```




```
<IMG SRC="alah.jpg" HEIGHT="300" WIDTH="500">
```



الخاصية التالية التي تستخدم مع <IMG> هي ALT وفيها نحدد نصاً بديلاً يظهر مكان الصورة. وهذا النص يلاحظ خصوصاً عندما يكون خيار "إظهار الصور تلقائياً" غير فعال في المتصفح. كما تستطيع ملاحظته في الفترة التي تسبق تحميل الصور وخاصة في المواقع بطيئة التحميل.

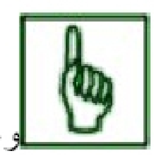
```
<IMG SRC="thedome.jpg" ALT="The Dome Of The Rock">
```

عندما نقوم بإدراج صورة ضمن فقرة فإن موقع ظهورها يتحدد بالطبع حسب ترتيب ورودها في الفقرة، مثلها مثل أي كلمة أو عبارة أخرى. ونستخدم الخاصية ALIGN لتحديد محاذاة الصورة مع النص المرافق لها أو لنقل بعبارة أخرى: تحديد موقع النص الذي يليها بالنسبة لها وهي تأخذ القيم: BOTTOM, TOP, MIDDLE, LEFT، RIGHT وأوضح لك تأثير كل قيمة كما يلي:

في الحالة العادية (مثل هذه) وعندما لا نقوم بتحديد أي محاذاة فإن النص الذي يلي الصورة يظهر بمحاذاة الحافة السفلى لها. وهذه هي الحالة الإفتراضية لظهور الصور والتي تمثلها القيمة BOTTOM

```
<IMG SRC="image.jpg" ALIGN="BOTTOM">
```

<u>TOP</u>



وعند تحديد هذه القيمة فإن السطر الأول من النص الذي يلي الصورة يقع بمحاذاة الحافة العليا لها. أما باقي النص فيمتد أسفلها.

```
<IMG SRC="image.jpg" ALIGN="TOP">
```

<u>MIDDLE</u>

أما عند تحديد هذه القيمة فإن السطر الأول من النص يقع بمحاذاة منتصف الصورة. كذلك فإن باقي النص يمتد أسفلها.

```
<IMG SRC="image.jpg" ALIGN="MIDDLE">
```

LEFT



هذه القيمة تؤدي إلى محاذاة الصورة إلى أقصى اليسار. مع التفاف النص الذي يليها على الجهة اليمنى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="LEFT">

RIGHT



أما هذه القيمة فتؤدي إلى محاذاة الصورة إلى أقصى اليمين. مع التفاف النص الذي يليها على الجهة اليسرى ولعدة أسطر حسب ارتفاع الصورة.

<IMG SRC="image.jpg" ALIGN="RIGHT">

والآن بعد أن قمنا بتحديد محاذاة الصورة نحتاج إلى تحديد المسافة الفاصلة بينها وبين النص الذي يجاورها. ونستخدم لذلك الخصائص التالية:

VSPACE: لتحديد المسافة العمودية الفاصلة بين النص والحافتين العليا والسفلى للصورة.

HSPACE: لتحديد المسافة الأفقية الفاصلة بين النص والحافتين اليمنى واليسرى للصورة.

مثال :

<IMG SRC="alah.jpg" ALIGN="RIGHT" VSPACE="20" HSPACE="20">



النتيجة: هذه الشيفرة ستدرج الصورة المسماه image.jpg مع محاذاتها ليمين الصفحة وإضافة مسافة فارغة مقدارها 20 بيكسل على الجهات الأربعة. (قارن بين هذا الإطار والإطار السابق الذي وضحت فيه خاصية ALIGN مع القيمة RIGHT. ولاحظ المسافة بين الصورة والنص المرافق لها.)

الخاصية الأخيرة والتي تستخدم مع الوسم <IMG> هي BORDER ووظيفتها إضافة إطار حول الصور والتحكم بسُمكِه. وهذه الخاصية تستخدم بشكل خاص عند تعيين صورة ما كوصلة تشعبية. ويتم التحكم بالسُمك من خلال إسناد رقم يمثل السُمك بالبيكسل. والقيمة الإفتراضية له هي 0 أي لا يوجد إطار حول الصورة.

مثلاً لإضافة إطار سُمكه 5 بيكسل نكتب الشيفرة التالية:

```
<IMG SRC="image.jpg" BORDER="5">
```

والآن حان الوقت لكي نناقش معاً بعض الأمور التي تتعلق بالصور والرسومات بشكل عام.

- **هل حاولت أن تتعرف على أنواع الملفات الرسومية التي تقوم بتحميلها خلال تصفحك لمواقع الإنترنت؟**

يزخر عالم الكمبيوتر بالعشرات من أنواع الملفات الرسومية وتنسيقات الصور. وكل منها يختلف عن غيره من عدة نواح، أذكر لك منها: الدقة، وعدد الألوان التي يستوعبها، والحجم التخزيني للملف. لكن هناك نوعين فقط من هذه الملفات يتم تداولهما حالياً في الإنترنت وهما:

JPG, JPEG

إختصار لِـ <u>J</u>oint <u>P</u>hotographic <u>E</u>xperts <u>G</u>roup. ويدعم هذا التنسيق صوراً بعيار 24 بت (أي 16.7 مليون لون). وميزة هذا التنسيق تتمثل في إمكانية ضغط الصور بنسب مختلفة عند تخزينها وبالتالي الحصول على صور صغيرة الحجم نسبياً.(أعني هنا حجم التخزين بالكيلوبايتات وليس أبعاد الصورة). لكن بالمقابل كلما إزدادت نسبة الضغط وصغر حجم الملف كان ذلك على حساب الجودة والوضوح.

GIF

إختصار لِـ <u>G</u>raphical <u>I</u>nterchange <u>F</u>ormat وأقصى عدد للألوان في هذا التنسيق هو 265 لون. ومع ذلك فإن أحجام الصور المخزنة به كبير نسبياً مقارنة بتنسيق JPG. لكن هناك مزايا رائعة ينفرد بها تنسيق GIF مما يستدعي استخدامه في صفحات الويب، أولها القدرة على تخزين صور بخلفيات شفافة Transparent Images وثانيها الصور المتحركة Animated Gifs

وتجد معلومات وافية ودروساً مفصلة حول هذه المواضيع ضمن دروس Paint Shop Pro.

والآن قد تسأل، أي من هذين التنسيقين أستخدم في صفحاتي؟! لا يوجد جواب قطعي لهذا السؤال لكن إليك هاتين المعادلتين:

JPG= الصور الحقيقية ذات العدد الكبير من الألوان، وذات الأبعاد الكبيرة
GIF= الصور قليلة الألوان وصغيرة الأبعاد مثل الأزرار.

ü ما هي درجة إستبانة شاشتك Resolution؟ إذا كنت لا تعرف الجواب قم بفتح تطبيق لوحة التحكم في ويندوز 95 وإختر أيقونة(العرض) ثم اختر التبويب (إعدادات) وهناك سوف تشاهد "مساحة سطح المكتب" الذي يدل على درجة إستبانة الشاشة، وعلى الأغلب ستكون 640×480 أو 800×600، وهناك درجات أعلى تعتمد على قدرة محول العرض. كذلك سوف تشاهد "لوح الألوان" الذي يدل على عدد الألوان التي يمكن عرضها بالإعدادات الحالية للشاشة.

أما في ويندوز 3.11 أو 3.1 فاختر أيقونة برنامج إعداد Windows من لوحة التحكم فتظهر لك قائمة تجد بضمنها نوع وإستبانة الشاشة.

هذا الحديث يقودني إلى وحدة البيكسل Pixel (ألم أعدك مسبقاً بتوضيحها). وهي اختصار لِـ Picture Element. إذا كانت شاشتك بإستبانة 640×480 فهذا يعني أنها مقسمة(نظرياً) إلى شبكة من 640 عمود و480 سطر. وبمنتهى البساطة، إن كل خلية من هذه الشبكة تمثل بيكسل وبالطبع كلما زادت الإستبانة كلما صغر حجم وحدة البيكسل.

ü هل سبق لك وأن سمعت بمصطلح Thumbnail ضمن مصطلحات الإنترنت؟ حسناً، لا تلتفت إلى الترجمة الحرفية لهذه الكلمة، والتي تعني "ظفر الإبهام".

فالمقصود حقيقةً بها هي تلك الصورة الصغيرة جداً التي تقوم بالنقر عليها فتؤدي إلى عرض صورة بحجم أكبر. لذلك قد يكون المصطلح الأنسب لوصفها هو "العيّنة".

(وإذا كنت قد زرت أحد المواقع الإخبارية لرأيت كيف يتم عرض عينات وصور مصغرة للقطات الأحداث وعند النقر على العينة تظهر الصورة الأصلية. إذن أنت لست مجبراً على الإنتظار لوقت طويل لحين ظهور صورة ذات حجم كبير للقطة لست معنياً بها).

ومن الواضح أن استخدام العينات مفيد وعملي جداً وأن وضعها في المواقع التي تحتوي على العديد من الصور يؤدي إلى تقليل الزمن اللازم لتحميل الصفحات وتجنب ضياع الوقت بانتظار ظهور الصور الأصلية كبيرة الحجم. لأنها تعطي الزائر الحرية في النقر عليها إذا رغب في رؤية الأصل أو تجاهلها. أما كيف يتم عمل هذه العينات؟ فذلك باستخدام أحد برامج معالجة الرسوم كبرنامج Pro Paint Shop. من خلال تصغير أبعاد الصور الأصلية إلى النسبة المطلوبة.

أعرف ماذا ستسأل الآن، ستقول ألم نتعلم قبل قليل كيفية عرض الصور مع التحكم بأبعادها؟ ألا يؤدي استخدام الخصائص WIDTH، HEIGHT إلى التحكم بحجم الصور وعرضها بنسب مصغرة حسب ما هو مطلوب؟

إن استخدامك لهذه الخصائص يؤدي إلى إظهار الصورة بحيث تبدو مصغرة، لكنك فعلياً قمت بإجبار متصفح الزائر على تحميل الصورة بالحجم والأبعاد الأصلية ثم عرضها بالحجم المصغر أي أنك في النهاية لم تحقق الغاية من وجود هذه العينات.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HH

الدرس السادس

الوصلات التشعبية

أهلاً وسهلاً بك إلى الدرس السادس من كتاب تعلم HTML.

Links... أو الوصلات التشعبية هي روح الإنترنت. وإذا كانت الإنترنت بمجملها هي شبكة العنكبوت فإن هذه الوصلات هي الخيوط التي تشكل هذه الشبكة وتؤلف حلقات الوصل بين الملايين من مواقعها. تنقر على وصلة ما فتنقلك إلى صفحة أخرى في نفس الموقع... وتنقر على وصلة أخرى لتنقلك كلياً إلى أحد المواقع في الجانب الآخر من العالم... وصلة تجعلك تحمّل ملفاً وأخرى تجعلك تشغّل مقطعا موسيقياً وثالثة تعرض لك صورة...

حسناً، من المؤكد أنك استنتجت الآن من هذه المقدمة أنك بصدد تعلم كيفية إدراج الوصلات التشعبية في صفحاتك... لقد صدق استنتاجك لذلك هيا إلى العمل...

هناك عدة خيارات للوصلات التشعبية، منها أن تكون الوصلة لموقع آخر، أو أن تكون لصفحة أخرى داخل الموقع نفسه، ومنها أن تكون لمكان آخر في نفس الصفحة (إلى أعلى أو أسفل على سبيل المثال) أو أن تكون وصلة لعنوان بريد إلكتروني E-mail وفي جميع الحالات فإن المبدأ واحد لكن تختلف بعض التفاصيل. وسوف أناقش معك كل حالة على حدة وبالتفصيل.

نستخدم الوسوم

</A> ... <A>

كوسوم أساسية لإدراج الوصلات التشعبية، وهي اختصار لكلمة Anchor. وهي لا تعمل لوحدها بل تتطلب إضافة خصائص معينة أولها وأهمها الخاصية

HREF

التي نحدد من خلالها الموقع الذي نريد الدلالة عليه، ويجب أن يكتب عنوان الموقع كاملاً.

الحالة الأولى: إدراج وصلة تشعبية تشير إلى موقع خارجي.

لنقم بإدراج وصلة تشعبية إلى أحد المواقع العربية الرائدة والرائعة، وهو موقع شركة صخر. وعنوانه http://www.Dinaro.com/vb3 في هذه الحالة يتم كتابة الشيفرة بالشكل التالي:

<A HREF="http://www.Dinaro.com/vb3"> </A>

لكن بقي شيء واحد وهو العبارة أو الكلمة التي سيتم النقر عليها لتشغيل الوصلة، وهذه يجب أن توضع بين الوسمين <A> ... </A>. أي لكي تكتمل الوصلة السابقة يجب أن نكتب معها أي عبارة نريدها، لكي ينقر عليها الزائر فتنقله إلى العنوان المطلوب. ما رأيك بعبارة: Go To Dinaro والتي تصبح الشيفرة معها بالشكل التالي:

<A HREF="http://www.Dinaro.com/vb3">Go To Dinaro </A>

وتظهر الوصلة كما يلي:

Go To Dinaro

لم تعجبك؟ ليس ذلك مشكلة فأنت تستطيع كتابة أي شيء تريده كعنوان للوصلة التي تريدها. ما رأيك لو جعلنا كلمة Dinaro هي فقط العنوان لهذه الوصلة

Go To <A HREF="http://www.Dinaro.com/vb3"> Dinaro </A>

Go To Dinaro

بل إنك تستطيع إدراج صورة أو (زر) كبديل عن الكلمات -كما تشاهد في الكثير من المواقع- وكل ما عليك فعله في هذه الحالة هو كتابة الوسم الخاص بإدراج الصورة بين الوسمين <A> ... </A> بالشكل التالي:

<A HREF="http://www.Dinaro.com/vb3""><IMG SRC=" DINARO1.gif"></A>

والذي يؤدي إلى ظهور الصورة التالية كوصلة تشعبية لموقع ديناروا

وبشكل عام فإن أي شيء يوضع بين الوسمين <A> ... </A> سوف يكون الوسيلة أو العنوان الذي ينقلنا إلى الموقع المشار إليه في الوصلة التشعبية، سواءً كان هذا الشيء نصاً أو صورة أو كلاهما معاً.

والآن هل تلاحظ الإطار الظاهر حول الصورة؟ وهل تذكر متى قمنا بالحديث عن هذا النوع من الإطارات؟ نعم، في الدرس السابق. عند إدراج صورة كوصلة تشعبية يظهر حولها إطار سمكه 2 بيكسل وهذه هي الحالة الإفتراضية. وبالطبع نستطيع إزالته بكتابة الخاصية "BORDER="0 ضمن وسم الصورة.

```
<A HREF="http://www.Dinaro.com/vb3""><IMG SRC="
DINARO1.gif" BORDER="0"></A>
```

أو حتى تكبيره بكتابة السمك المطلوب مع هذه الخاصية.

```
<A HREF="http://www.Dinaro.com/vb3""><IMG SRC="
DINARO1.gif" BORDER="7"></A>
```

ننتقل الآن إلى الحالة الثانية، وهي أن تشير الوصلة التشعبية إلى ملف موجود في نفس الموقع (أي ملف محلي) سواءً كان ملف HTML أو صورة أو غير ذلك. وفي هذه الحالة فإن ما يكتب مع الخاصية HREF هو اسم هذا الملف المطلوب الوصول إليه.

لنقم بإنشاء وصلة تشعبية تقودنا إلى الصفحة الرئيسية لهذا الموقع وبالمناسبة فإن الملف الذي يحتويها اسمه index.html ، والشيفرة الخاصة بذلك هي:

```
<A HREF="index.html">Main Page</A>
```

Main Page

وأذكرك بأنك تستطيع إدراج صورة هنا أيضاً كعنوان للوصلة التشعبية وذلك بنفس التفاصيل التي شرحتها في الحالة السابقة.

هيا ندرج صورة مصغرة كعنوان لوصلة تشعبية للصورة الأصلية.

```
<A HREF=" image.jpg"><IMG SRC=" image _1.jpg"
BORDER="0"></A>
```

في هذا المثال قمت بتحديد الصورة المصغرة المسماه image _1.jpg كوصلة تشعبية تصلنا إلى الصورة الأصلية المسماه image.jpg



لكن إنتبه إذا كان الملف المطلوب والذي تريد الإشارة إليه موجوداً في مجلد مختلف عن المجلد الذي يوجد به الملف الحالي، فيجب عليك تحديد المسار الكامل لهذا الملف وذلك بنفس الطريقة التي ناقشناها في الدرس السابق عندما قمنا بإدراج الصور.

الحالة الثالثة هي أن نقوم بالإشارة إلى مكان آخر داخل نفس الصفحة، إلى أولها مثلاً أو إلى آخرها أو أي مكان آخر نريده...

طبعاً مهما بلغت درجة الذكاء والألمعية التي يتصف بها الكمبيوتر ومتصفح الإنترنت، فهما لا يستطيعان معرفة ما يدور بفكرك وبالتالي لا يستطيعان معرفة المكان الموجود في نفس الصفحة والذي تريد نقل زائرك إليه من خلال الوصلة التشعبية. لذلك يجب أن تقوم أنت بتحديده.

والمبدأ هنا هو أن تقوم بتعريف أو تسمية هذا المكان بإسم معين سوف تقوم لاحقا باستخدامه في الوصلة التشعبية. وفي هذه الحالة يتحتم عليك استخدام الخاصية الثانية للوسم <A> وهي NAME

لنقم معاً بإدراج وصلة تشعبية داخل هذه الصفحة تقوم بنقل الزائر من مكان وجود هذه الوصلة إلى الفقرة الثالثة من هذه الصفحة والتي بدأنا فيها الحديث عن الوصلات التشعبية Links

أول ما يجب فعله هو الذهاب إلى هذه الفقرة واختيار أول كلمة فيها ثم وضعها داخل الوسوم <A> ... </A>

```
<A>LINKS</A>
```

والآن حان الوقت لاستخدام الخاصية NAME فالخطوة الثانية هي تعريف هذه الكلمة بأي اسم نريده (المهم أن نبقى متذكرين له). سوف أقوم بإعطاء الاسم attrib1

```
<A NAME="attrib1">LINKS</A>
```

لقد أصبحت هذه الفقرة جاهزة لكي نقوم بإدراج وصلات تشعبية إليها من أي مكان في هذا الملف، بل ومن أي ملف آخر ... وأكثر من ذلك أنه إذا أراد أحد ما في أحد المواقع الأخرى أن يضع وصلة تشعبية لها من موقعه فإن باستطاعته ذلك شرط أن يعرف الإسم الذي عرّفناها به وهذا ليس صعباً بالطبع.

الخطوة الثالثة هي إدراج الوصلة التشعبية لهذه الفقرة.

ويلزمنا هنا معرفة اسم الملف الذي توجد به هذه الفقرة (أي هذا الملف الذي نعمل به) واسمه htutor06.html لأنه سيشكل المدخل الأساسي للوصول إلى الفقرة المحددة. وتكون شيفرة الوصول إلى هذه الفقرة هي كالتالي:

```
<A HREF="htutor06.html#attrib1">3rd Paragraph</A>
```

3rd Paragraph

لاحظ أننا لم نكتف بذكر اسم الفقرة لوحدها بل يجب أن تقرن باسم الملف الأب الذي يتضمنها من خلال إشارة #

أما الحالة الأخيرة والتي نقوم فيها بإدراج وصلة تشعبية لعنوان بريد إلكتروني، يؤدي النقر عليها إلى إطلاق برنامج البريد الإلكتروني للزائر بشكل تلقائي.

فالإختلاف الوحيد الذي يطرأ هنا هو كتابة كلمة DINARO بعد خاصية HREF لكي تدل على أن العنوان الذي يلي هو عنوان GMAIL وليس أي عنوان آخر

<A HREF="DINARO:DINARO2AYOUB@GMAIL.COM">
GMAIL</A>

GMAIL

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHH

الدرس السابع

الجداول [1]

أهلاً وسهلاً بك إلى الدرس السابع من كتاب تعلم HTML.

هذا الدرس سيكون الأول من درسين حول الجداول. وقد فضلت تجزئتها إلى قسمين وذلك لأهمية هذا الموضوع وتعدد خصائص الجداول واحتمالات استخدامها في صفحات الويب.

تعد الجداول من أقوى الأدوات التي تتضمنها لغة HTML وأكاد أجزم بأنه لا يوجد موقع في الإنترنت إلا ويستخدمها بصورة أو بأخرى. والحقيقة أن وضع الجداول في صفحات الويب لا يقتصر على تلك القوائم من البيانات التي نحتاج لترتيبها في صفوف وأعمدة، بل يتعدى ذلك إلى استخدامها في تصميم الصفحات نفسها وتنظيمها، والتحكم بمظهرها بصورة قوية وفعالة لا يمكن أداؤها مهما استخدمنا من وسوم خاصة بتنسيق الصفحات.

إن التعامل مع الجداول وإدراجها في صفحات الويب سهل جداً مثله مثل أي أداة من أدوات HTML لكنه قد يبدو لك مربكاً بعض الشيء وخاصة في البداية، وذلك لتعدد الخصائص التي تستعمل معها وتعدد الأوجه التي نستطيع التصرف بها. لكن لا تقلق فكل شيء يبدو صعباً في بدايته ولكن سرعان ما يصبح سهلاً.

هل أنت مستعد يا عزيزي المتدرب؟ إذن هيّا بنا...

بداية، إليك هذا الوصف البسيط للوسوم الأساسية الخاصة بالجداول

<TABLE>...</TABLE>	وسوم تعريف الجدول
<TR>...</TR>	Table Row وسوم تعريف الصف في الجدول
<TD> *Cell Data* </TD>	Table Data وسوم تعريف الخلايا في الصف وتعريف محتويات كل خلية

والآن لنتكلم بصورة أكثر دقة وتفصيلاً:

هذه هي الوسوم التي نبدأ بها لإدراج جدول مكون من خلية واحدة أو من مليون خلية... الأمر سيان

</TABLE> ... <TABLE>

والآن بعد إدراج هذين الوسمين، هناك سؤالين ينبغي الإجابة عليهما. الأول: كم عدد الصفوف التي نريدها في الجدول؟ ثلاثة، أربعة، مائة؟ لا بأس، قم بإضافة الوسوم

</TR> ... <TR>

بنفس عدد الصفوف التي تريدها. ولنفترض هنا أنها ثلاثة.

<TABLE>

<TR>
</TR>

<TR>
</TR>

<TR>
</TR>

</TABLE>

والسؤال الثاني هو، كم عدد الخلايا (أو الأعمدة) التي نريدها في كل صف؟

</TD> ... <TD>

بنفس عدد الخلايا المطلوب. ومن البديهي أن نكتبها بين الوسوم <TR> ... </TR> طالما أن الخلايا هي جزء من الصفوف. وهنا سأفترض أننا نريد خليتين في كل صف، وبذلك يجب تكرار كتابتها مرتين لكل صف.

وأذكرك أن النص الذي نريد إدراجه في الخلية يكتب ضمن هذين الوسمين

<TABLE>

<TR>

<TD> *Data* </TD>
<TD> *Data* </TD>

```
</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

هل اتضحت لك الصورة الآن. أنظر إلى نتيجة العمل التي حصلنا عليها.

Data	Data
Data	Data
Data	Data

هناك شيء ما ينقص. بالطبع ... الحدود. انتظر قليلاً وستعرف الخاصية التي تقوم بإضافة الحدود للجدول وغيرها من الخصائص الأخرى. لأنني قبل أن أستمر أود أن ألفت نظرك لمسألة معينة في الجداول. وهي أن طريقة التعامل معها تتم على ثلاثة مستويات:

- مستوى الجدول ككل
- مستوى الصفوف ككل أو كل واحد على حده
- مستوى الخلايا ككل أو كل واحدة على حده

ولكل من هذه المستويات خصائصه التي ينفرد بها كما أن هناك خصائص مشتركة تستخدم مع كل الوسوم.

نبدأ بمناقشة الخصائص التي تستخدم مع الوسوم <TABLE> ... </TABLE> وسأقوم أولاً بسردها لك، ومن ثم إدراج بعض الأمثلة التي توضحها.

BORDER	تقوم هذه الخاصية بإضافة حدود للجدول وتحديد سماكتها، والقيمة الإفتراضية لها هي صفر أي لا حدود <TABLE BORDER="5"> <TABLE BORDER="0">
WIDTH	تستخدم هذه الخاصية لتحديد عرض الجدول ككل. وهناك أسلوبين لتحديد العرض: المطلق أي بكتابة الرقم الذي يمثل العرض بصورة مباشرة. والنسبي أي كتابة رقم نسبي مئوي يحدد عرض الجدول حسب عرض نافذة المتصفح. (أي أن عرض الجدول سيختلف باختلاف عرض نافذة المتصفح). <TABLE WIDTH="600"> <TABLE WIDTH="80%">
HEIGHT	لتحديد ارتفاع الجدول، ويكون تحديد هذا الإرتفاع من خلال قيمة مطلقة تحدد الإرتفاع بالبيكسل. أو قيمة نسبية تحدد ارتفاع الجدول بالنسبة لإرتفاع صفحة المتصفح <TABLE HEIGHT="500"> <TABLE HEIGHT="100%">
CELLSPACING	لتحديد المسافة بين كل خلية من خلايا الجدول <TABLE CELLSPACING="10">
CELLPADDING	لتحديد المسافة الفاصلة بين الحدود وبداية النص في كل خلية. أو لنقل: تحديد حجم الهوامش لخلايا الجدول. <TABLE CELLPADDING="10">

ALIGN	لتحديد محاذاة الجدول أفقياً على الصفحة يميناً أو يساراً. وهو يأخذ القيم right, left <TABLE ALIGN="Left"> <TABLE ALIGN="Right">
BGCOLOR	ويستخدم لتحديد لون الخلفية للجدول <TABLE BGCOLOR="#00FFFF">

هذه هي الخصائص المستعملة مع الجدول. وسأقوم الآن بتطبيقها على المثال الوارد في بداية هذا الدرس وسأكتفي بكتابة وسم البداية أما باقي الوسوم فهي نفسها:

<TABLE BORDER="5">

Data	Data
Data	Data
Data	Data

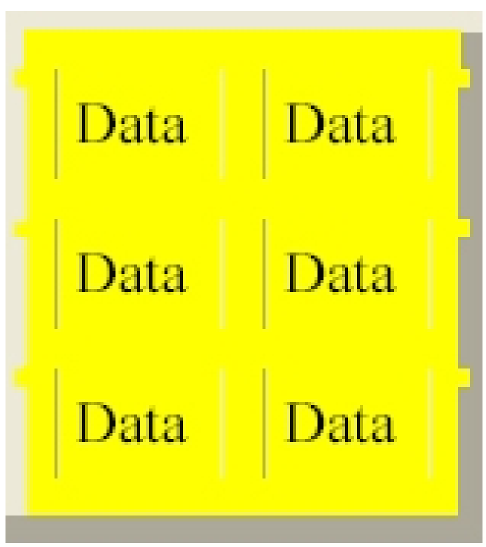
<TABLE BORDER="5" CELLPADDING="5">

Data	Data
Data	Data
Data	Data

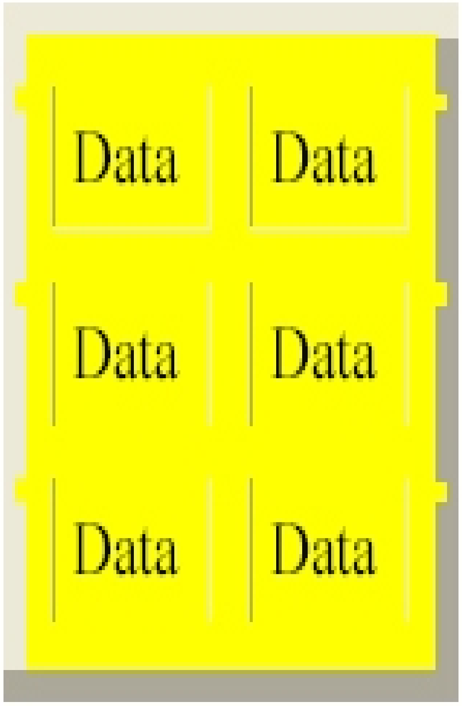
<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10">



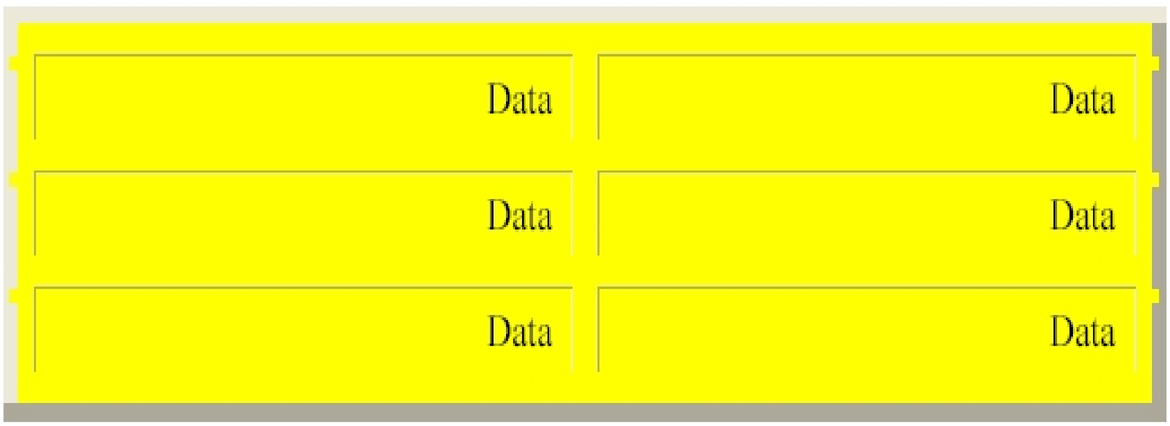
<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00">



<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00" HEIGHT="300">



<TABLE BORDER="5" CELLPADDING="5" CELLSPACING="10" BGCOLOR="#FFFF00" HEIGHT="300" WIDTH="75%">



ونتكلم الآن عن الخصائص المستخدمة مع وسوم الصف <TR> ... </TR> ولا بأس من تذكيرك أن عدد الصفوف في الجدول يتحدد بعدد هذه الوسوم. أما أهم الخصائص التي تضاف لهذا الوسم فهي:

ALIGN	لتحديد محاذاة النص أفقياً داخل الخلايا التي يتكون منها الصف، والقيم المحتملة لها هي Right, Left, Center والقيمة الإفتراضية هي Center
VALIGN	لتحديد المحاذاة العمودية للنص داخل خلايا الصف، وذلك إما للأعلى أو للأسفل أو في المنتصف أو على امتداد الخط الأساسي للخلية. وقيمها على التوالي هي: Baseline ,Top, Bottom, Middle
BGCOLOR	لتحديد لون الخلفية للخلايا التي يتكون منها الصف. وهنا يتم تجاهل لون الخلفية المحدد ضمن وسم <TABLE> ويتم تطبيق اللون المحدد هنا.

ونعود الآن إلى جدولنا السابق لنطبق عليه هذه الخصائص من خلال الأمثلة التالية:

```
<TABLE BORDER="5" HEIGHT="300">
<TR ALIGN="Left">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Right">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR ALIGN="Center">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

```
<TABLE BORDER="5" HEIGHT="300">
<TR VALIGN="Top">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR VALIGN="Bottom">
<TD> Data </TD>
<TD> Data </TD>
</TR>

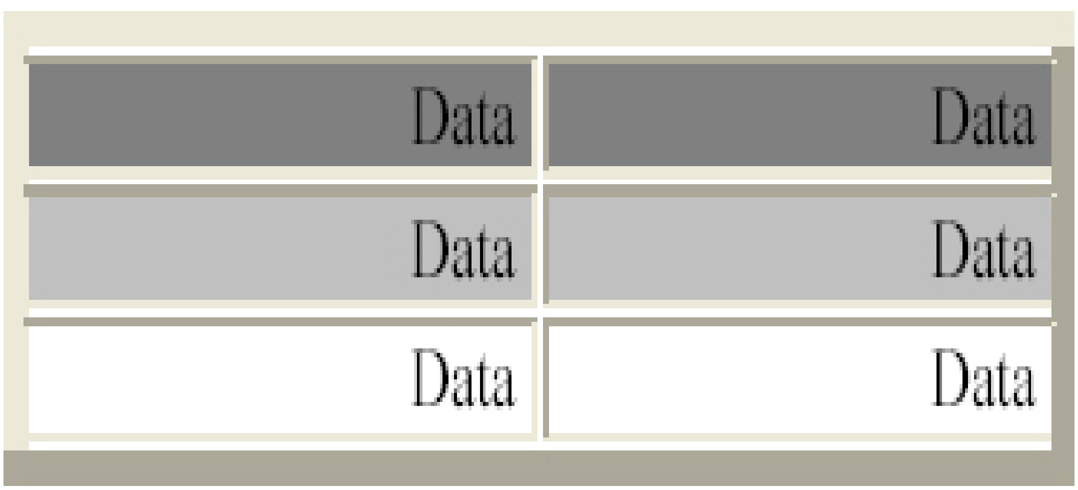
<TR VALIGN="Baseline">
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```

Data	Data
Data	Data
Data	Data

```
<TABLE BORDER="5" HEIGHT="300"
BGCOLOR="#FFFFFF">
<TR BGCOLOR="#808080">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR BGCOLOR="#C0C0C0">
<TD> Data </TD>
<TD> Data </TD>
</TR>

<TR>
<TD> Data </TD>
<TD> Data </TD>
</TR>
</TABLE>
```



Data	Data
Data	Data
Data	Data

والآن ماذا بقي لدينا؟ بالطبع بقيت الوسوم الخاصة بالخلايا. وسوف أتابع الحديث عنها في الدرس القادم إن شاء الله. أراك في الدرس التالي لنتابع الحديث عن موضوع الجداول.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHH

الدرس الثامن

الجداول [2]

أهلاً وسهلاً بك إلى الدرس الثامن من كتاب تعلم HTML.

نتابع معاً في هذا الدرس الحديث عن الجداول. وأنا أفترض أنك قد أنهيت الدرس السابق بنجاح، وأن لديك الآن فكرة جيدة جداً عن الجداول وكيفية إنشائها والتعامل مع خصائصها ومع الصفوف وخصائصها. ونكمل الآن من حيث توقفنا، أي مع خصائص الخلايا.

هل تذكر ما قلناه عن عدد الخلايا في الصف الواحد؟ إن عدد الخلايا المطلوب يتحدد من خلال كتابة الوسوم <TD> ... </TD> مرات بنفس العدد المطلوب. ومن الممكن أن تحتوي الخلية على أي عنصر من عناصر لغة HTML : نصوص، رسوم، قوائم، وصلات تشعبية، بل وحتى جداول. (نعم، تستطيع إدراج جدول داخل جدول آخر)

لنسترجع معا المثال الذي قمنا بالتدرب عليه في الدرس السابق، فسوف نكمل هذا الدرس معه. وهو جدول صغير مكون من ثلاثة صفوف وعمودين (أي خليتين في كل صف).

```
<TABLE>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

أما الخصائص المستخدمة مع الخلايا، فهذا جدول بها:

ALIGN	تحدد محاذاة النص الموجود في الخلية أفقياً، والقيم المستخدمة هي Left, Center, Right
VALIGN	تحدد المحاذاة العمودية للنص، وهو يأخذ القيم Top, Middle, Bottom, Baseline
WIDTH	تحدد عرض الخلية، وذلك بكتابة القيمة المباشرة للعرض المطلوب بالبيكسل، أو بكتابة رقم يمثل النسبة المئوية. ويكفي تحديد العرض للخلايا في أحد الصفوف لكي يتم تطبيقه على كل الخلايا في كل الصفوف.
HEIGHT	تحدد الإرتفاع المطلوب للخلية في الصف، وذلك بالطرق المباشرة أو النسبية. وقيامك بتحديد ارتفاع إحدى الخلايا في الصف يؤدي إلى تطبيقه على كل الخلايا فيه.
BGCOLOR	تحدد لون خلفية الخلية
COLSPAN	يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها أفقياً <TD COLSPAN="n"> حيث n هو عدد الخلايا التي سيتم دمجها
ROWSPAN	يقوم بدمج الخلية الحالية مع العدد المطلوب من الخلايا التي تليها عمودياً (أي أسفلها). <TD ROWSPAN="n"> وبالطبع n هو عدد الخلايا التي سيتم دمجها

وقبل أن نستمر، يبدو لي أن هناك بعض الملاحظات المهمة التي ينبغي ذكرها:

كما تلاحظ هناك خصائص تتكرر مع جميع الوسوم. خذ مثلاً الخاصية BGCOLOR. كيف يتم التعامل معها إذا كررت مع جميع الوسوم؟ بكل بساطة يتم تطبيق اللون المحدد مع وسم الخلية، فإذا لم يكن محدداً يطبق اللون المحدد مع وسم الصف، فإذا لم يوجد يطبق اللون المحدد مع وسم الجدول. وإذا لم يكن هذا محدداً بدوره يتم إعتماد لون خلفية الصفحة المحدد في الوسم <BODY>.

الملاحظة الثانية تتعلق بالخصائص WIDTH، HEIGHT. يختلف أسلوب التعامل مع هذه الخصائص من متصفح لآخر، بل وتختلف أيضاً طريقة تفسير القيم المحددة معها وخصوصاً فيما يتعلق بالنسب المئوية. (راجع الموضوع: الوسوم الخاصة والمتصفحات).

وبدون الخوض في تفاصيل هذه الإختلافات التي لن تؤدي إلا إلى المزيد من الإشكالات لديك... وبعد التجربة يبدو أن أفضل طريقة للتعامل مع هذه الخصائص هي قيامك بتحديد العرض (وكذلك الإرتفاع إذا أردت ذلك) للجدول ككل من خلال الوسم <TABLE>. ثم استخدام هذه الخصائص في وسوم الخلايا وتحديد العرض المطلوب لكل خلية على حده في الصف الأول، والارتفاع المطلوب لكل صف في الجدول.

وهذه برأيي أفضل طريقة تضمن بها أفضل مشاهدة للجدول لجميع زوار موقعك.

إذا أردت أن تحتوي بعض الصفوف في الجدول على عدد من الخلايا أقل من باقي الصفوف، فلا يكفي أن تقوم بحذف وسوم الخلايا منها. (كما ترى في الشيفرة التالية:)

```
<TABLE BORDER="5">

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

لأن هذا ما ستحصل عليه:

	Data
Data	Data
	Data

لقد بقي مكان الخلايا المحذوفة محجوزاً كما لو أنها لم تحذف. أما الخلايا الباقية فظلت محتفظة بنفس خصائصها، أي أننا لم نستفد من عملية الحذف.

والحقيقة أن الطريقة المثلى لذلك هي أن تقوم بدمج الخلايا معاً وذلك باستخدام الخصائص COLSPAN، ROWSPAN.

إذن لنقم بإعادة كتابة شيفرة الجدول مع استخدام هذه الخصائص:

```
<TABLE BORDER="5">
<TR>
<TD COLSPAN="2"> Data </TD>
</TR>
<TR>
<TD> Data </TD>
<TD> Data </TD>
</TR>
<TR>
<TD COLSPAN="2"> Data </TD>
</TR>
</TABLE>
```

لأن هذا ما ستحصل عليه:

Data	
Data	Data
Data	

لاحظ أن العدد 2 هو عدد الخلايا التي قمنا بدمجها. ولاحظ أيضاً انني لم أقم بإعادة وسوم الخلايا المحذوفة لأننا أصلاً لا نحتاج لها بعد أن قمنا بالدمج.

وكقاعدة أساسية: كل خلية يتم دمجها يجب بالمقابل حذف وسوم التعريف الخاصة بها. ما عدا تعريف الخلية الأساسية بالطبع.

مثال آخر: لنقم بدمج الخلايا الموجودة في العمود الأول

```
<TABLE BORDER="5">

<TR>

<TD ROWSPAN="3"> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
```

```
</TR>
<TR>
<TD> Data </TD>
</TR>
</TABLE>
```

ومرة أخرى بعد تعريف خاصية الدمج العمودي، قمت بحذف تعريف الخلايا المدموجة من الصف الثاني والثالث. وهذا هو الجدول الناتج.

<table>
<tr><td>Data</td><td rowspan="3">Data</td></tr>
<tr><td>Data</td></tr>
<tr><td>Data</td></tr>
</table>

هناك نوع خاص من الخلايا التي يتم تعريفها باستخدام الوسوم <TH> ... </TH> وهي اختصار Table Header أي ترويسة الجدول.

والفرق الوحيد بينها وبين <TD> ... </TD> هو أن النص الذي تحتويه يظهر بخط أسود عريض ومحاذاته في منتصف الخلية بصورة إفتراضية. (ليس بالشيء المهم، كما أعتقد)، خاصة وأن الخصائص المستخدمة معها هي نفس خصائص <TD> وبنفس التفاصيل التي ذكرت.

الوسوم الأخيرة المستخدمة في الجداول هي <CAPTION> ... </CAPTION> وهي تختص بإضافة عنوان رئيسي للجدول ككل. لذلك فهي عندما تكتب يتم وضعها

مباشرة بعد الوسم <TABLE> وبصورة مستقلة وليس ضمن وسوم الصفوف أو الخلايا.

```
<TABLE BORDER="5">
   <CAPTION> Table Caption </CAPTION>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

<TR>

<TD> Data </TD>
<TD> Data </TD>

</TR>

</TABLE>
```

Table Caption

Data	Data
Data	Data
Data	Data

وأخيراً... وصلنا إلى نـهاية هذه الدرس. وما بقي لدي هو أن أحثـك على تطبيق ما قمت بشرحه فيه وفي الدرس السابق وباقي الدروس، وتـجربة جميع الاحتمالات الواردة للخصائص والقيم.

أليس هذا ممتعا !!

HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH

الدرس التاسع

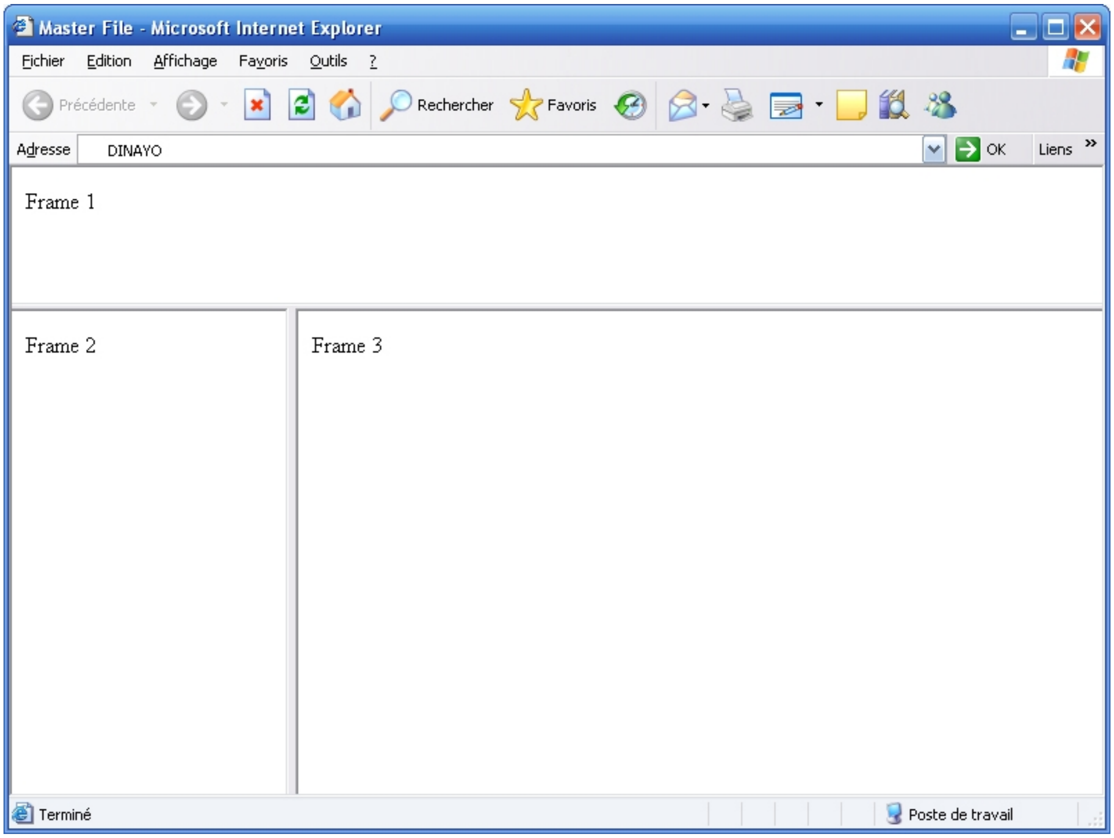
الإطارات

أهلاً وسهلاً بك إلى الدرس التاسع من كتاب تعلم HTML.

في هذا الدرس سوف نقوم بالتعرف على الإطارات Frames وطريقة عرض صفحات الويب باستخدامها...

فهل تعرف ما هي الإطارات؟ حسنا، سأوضحها لك... هل سبق لك وأن زرت إحدى الصفحات لتشاهد أنها مقسمة إلى عدة أقسام بحيث يظهر في كل منها صفحة مستقلة، وتبدو بصورة منفصلة عن الأقسام الأخرى. وربما تكون قد قمت بالنقر على إحدى الوصلات التشعبية الموجودة في أحد الأقسام لتظهر الصفحة المتعلقة بها في القسم الآخر.

إذا كنت قد شاهدت مثل هذه الصفحات فهذا يعني أن الإطارات مألوفة لديك وإلا فشاهد مثالاً على صفحة ذات إطارات



كما شاهدت، فإن الصفحة مكونة من ثلاثة أقسام: علوي وأيسر وأيمن... والحقيقة أن كل قسم منها هو عبارة عن ملف Html كامل ومستقل بحد ذاته. وهي مجرد صفحات عادية لا تختلف أبداً عن تلك التي تعلمت إنشاءها في الدروس السابقة، ولا علاقة لكل منها بالصفحات الأخرى من حيث التركيب والتعريف.

أما كيف تم جمعها معا لتظهر بالشكل الذي شاهدته؟ فهنا بيت القصيد. فبالإضافة إلى الصفحات والملفات الإعتيادية يوجد دائماً ملف أساسي يتم إنشاؤه خصيصا لتعريف صفحة الإطارات وتجميعها وتحديد خصائصها. أي أن المعادلة تتلخص بِـ:

مكونات صفحة الإطارات = عدد ملفات الصفحة نفسها + صفحة الملف الأساسي الذي يجمعها.

أي أنني في المثال السابق إحتجت فعلياً إلى أربعة ملفات لتكوين الصفحة.

وقبل أن نبدأ يا عزيزي المتدرب... لنقم بالتحضير للأمثلة التي سترد في هذا الدرس. لذلك قم بإنشاء ثلاثة أو أربعة ملفات بسيطة لكي تستخدمها في تطبيق الأمثلة أو استخدم ملفاتك القديمة التي قمت بالتدرب عليها في الدروس السابقة. أنا قمت بإنشاء ملفات على النمط التالي (وهي التي استخدمتها في المثال) وأسميتها frame1.html, frame2.html, frame3.html

```
<HTML>
<HEAD>
<TITLE>Frame1</TITLE>
</HEAD>
<BODY>
Frame 1
</BODY>
</HTML>
```

ونبدأ الآن بتعريف الملف الرئيسي الذي سيضم كافة الإطارات والملفات. وهو بالمناسبة ملف ذو حالة خاصة حيث نقوم باستخدام الوسوم

<FRAMESET> ... </FRAMESET>

بدلاً من الوسوم <BODY> ... </BODY>

((إذن الملف الرئيسي للإطارات لا يتضمن تعريفا باستخدام BODY))

```
<HTML>
<HEAD>
<TITLE>Master File</TITLE>
</HEAD>

<FRAMESET>
</FRAMESET>

</HTML>
```

نأتي الآن إلى الخصائص: والخاصية الأولى التي تستخدم مع هذه الوسوم هي COLS وهي تعرّف عدد وأحجام الإطارات العمودية للصفحة. وتُحدد الأحجام بطريقتين (هل عرفتهما؟) نعم... إنهما الطريقة المباشرة والطريقة النسبية...أو كلاهما معا.

والآن إليك هذه الأمثلة التي توضح مفهوم الأعمدة... وألفت نظرك إلى أن قيامك بالنقر على الشيفرة لكل مثال سيؤدي بك إلى مشاهدة هذا المثال بصورة عملية لكن إنتبه! فهذه الشيفرة غير مكتملة وكتابتها بهذا الشكل فقط لن يؤدي إلى أي نتيجة ولا إلى ظهور أي إطارات حيث ينقصها وسوم أخرى خاصة بمصدر الملفات الظاهرة داخل الإطارات، وقد قمت بإكمالها لغرض توضيح النتيجة لك فقط. لذلك أرجو أن تكتفي الآن بمعاينة كل شيفرة ونتيجتها إلى أن أقوم بسرد باقي الخصائص المهمة لاحقاً.

يحدد إطارين عموديين حجم كل منهما 50% من حجم الشاشة	<FRAMESET COLS="50%,50%"> </FRAMESET>
يحدد ثلاثة إطارات أحجامها 20% و 50% و 30% على التوالي من حجم الشاشة	<FRAMESET COLS="20%,50%,30%"> </FRAMESET>

<FRAMESET COLS="200,300,*"> </FRAMESET>	يحدد ثلاثة إطارات عمودية الأول حجمه 200 بيكسل، والثاني 300 بيكسل، أما الثالث * أي انه غير محدد بحجم معين ولكنه سيكون بالحجم المتبقي من الشاشة (طالما أننا لا نعرف استبانة الشاشة التي يستخدمها زائر الموقع)
<FRAMESET COLS="200,*,15%,20%"> </FRAMESET>	يحدد أربعة إطارات حجم الأول هو 200 بيكسل، والثالث 15% من حجم الشاشة، والرابع 20% من حجم الشاشة أما الثاني فسيكون حجمه بما تبقى من الشاشة.
<FRAMESET COLS="150,*,2*"> </FRAMESET>	يحدد ثلاثة إطارات الأول حجمه 150 بيكسل.... أما المساحة المتبقية فتقسم على أساس أن الإطار الثالث حجمه هو ضعفي (2*) حجم الإطار الثاني (*)

أما الخاصية الثانية فهي ROWS وأعتقد أنك استنتجت طبيعة عملها. نعم هي تحدد عدد وحجم الإطارات الأفقية (الصفوف) داخل الصفحة. وذلك بنفس الأسلوب المتبع مع الأعمدة، أي إما باستخدام الطريقة النسبية أو المطلقة. وسأقوم بسرد بعض الأمثلة لتوضيحها (وأذكرك ثانية أن هذه الأمثلة غير مكتملة):

<FRAMESET ROWS="50%,50%"> </FRAMESET>	يحدد إطارين أفقيين ارتفاع كل منهما 50% من ارتفاع الشاشة
<FRAMESET ROWS="20%,50%,30%"> </FRAMESET>	يحدد ثلاثة إطارات أفقية ارتفاعاتها 20% و 50% و 30% على التوالي من ارتفاع الشاشة
<FRAMESET ROWS="50,120,*"> </FRAMESET>	يحدد ثلاثة إطارات أفقية الأول ارتفاعه 50 بيكسل، والثاني 120 بيكسل، والثالث سيكون بالإرتفاع المتبقي من الشاشة
<FRAMESET ROWS="50,*,15%,20%"> </FRAMESET>	يحدد أربعة إطارات أفقية ارتفاع الأول هو 50 بيكسل، والثالث 15% من ارتفاع الشاشة، والرابع 20% من ارتفاع الشاشة أما الثاني فسيكون ارتفاعه بما تبقى من ارتفاع الشاشة.

يحدد إطارين الثاني ارتفاعه ضعفي ارتفاع الأول	<FRAMESET COLS="*,2*"> </FRAMESET>

لم ننته بعد من ذكر كل الخصائص المتعلقة بالوسوم <FRAMESET> فلا زال هناك الكثير. ولكن من الضروري أن نقوم الآن بالإنتقال إلى وسم آخر للإطارات لأنه مرتبط إرتباطاً وثيقاً بالوسوم السابقة وخصائصها المذكورة أعلاه، وهي <FRAME> فما هو عمل هذا الوسم؟

حسنا، كل ما قمنا به حتى الآن هو تعريف مجموعة من الإطارات وخصائصها (فقط تعريف الإطارات) لكن لم نحدد ماهية هذه الإطارات ولا محتوياتها ولا مصادرها. تماما كما نقوم بتعريف صفحات الويب الإعتيادية وخصائصها في الوسم <BODY> دون أن يعني ذلك تحديد محتويات هذه الصفحات. فإذا أردنا فيما بعد إدراج صورة مثلاً نستخدم الوسم الخاص بذلك وهو <IMG SRC="imagname.ext">

وفي حالة الإطارات فإننا نستخدم الوسم <FRAME> وهو وسم مفرد أي ليس له وسم نهاية تماماً مثل <IMG>. وفيه نقوم بتحديد مصدر وخصائص كل ملف نريد إظهاره داخل أحد الإطارات. ويتم استخدام هذا الوسم مرات بنفس عدد الإطارات المذكورة داخل <FRAMESET>. وسوف أقوم مباشرة باستخدام الخاصية SRC لتحديد مصدر الملف.

دعنا نقوم الآن بإتمام الشيفرة لبعض الأمثلة المذكورة أعلاه. ونبدأ بالمثال الأول:

```
<FRAMESET COLS="50%,50%">
   <FRAME SRC="frame1.html">
```

```
   <FRAME SRC="frame2.html">
</FRAMESET>
```

الآن ... والآن فقط أصبح لديك صفحة إطارات محترمة.

مثال آخر:

```
<FRAMESET COLS="200,400,*">
   <FRAME SRC="frame1.html">
   <FRAME SRC="frame2.html">
   <FRAME SRC="frame3.html">
</FRAMESET>
```

مثال ثالث:

```
<FRAMESET ROWS="50,*,15%,20%">
   <FRAME SRC="frame1.html">
   <FRAME SRC="frame2.html">
   <FRAME SRC="frame3.html">
   <FRAME SRC="frame4.html">
</FRAMESET>
```

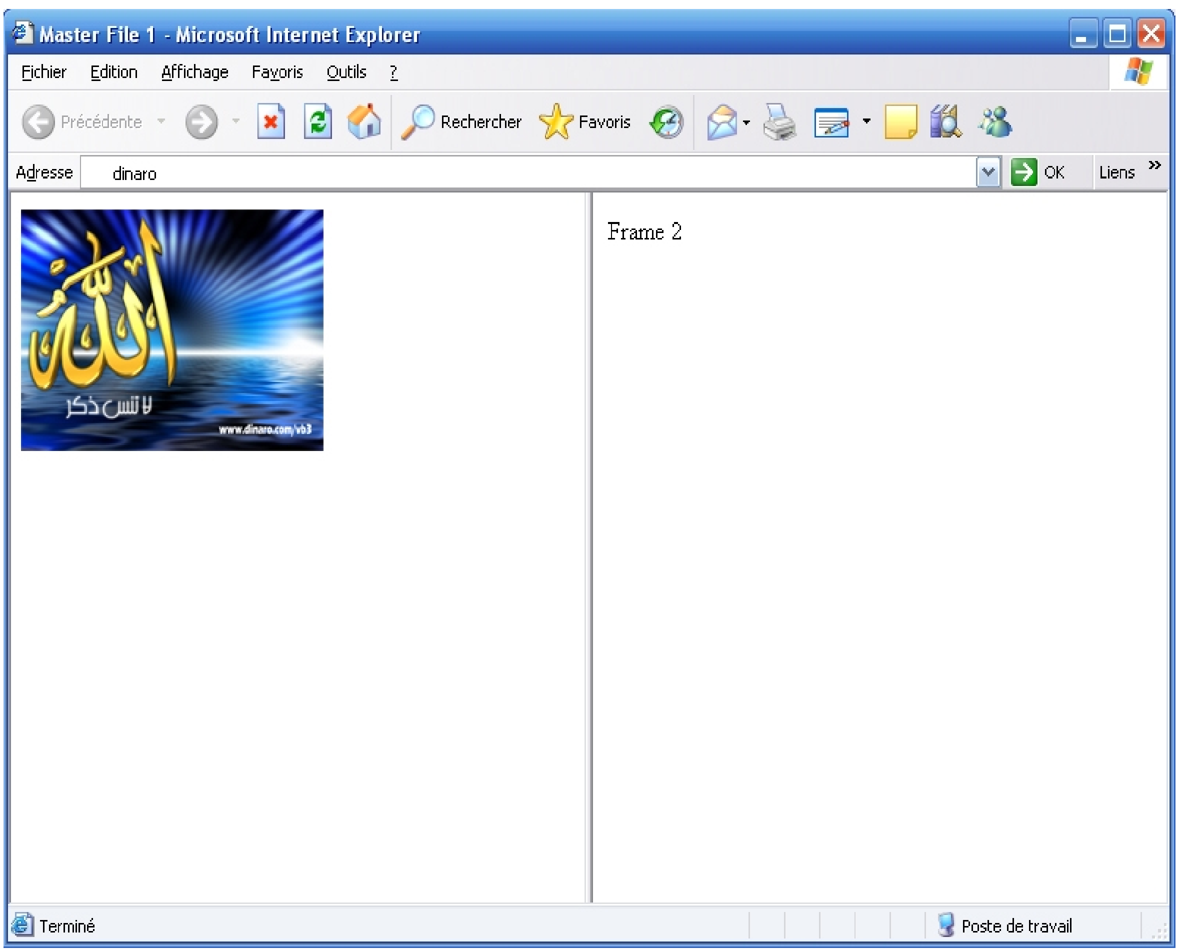
مثال رابع:

```
<FRAMESET COLS="*,2*">
   <FRAME SRC="frame1.html">
```

```
   <FRAME SRC="frame2.html">
</FRAMESET
```

وبالإضافة إلى ما ذكر، نستطيع إدراج صورة مباشرةً داخل الإطار وباستخدام <FRAME SRC> تماماً كما ندرجها باستخدام <IMG SRC> وإليك هذا المثال

```
<FRAMESET COLS="50%,50%">
   <FRAME SRC="frame1.html">
   <FRAME SRC="alah.jpg">
</FRAMESET>
```



والآن لماذا لا نقم معاً بمراجعة الأفكار الأساسية السابقة الذكر وتلخيصها؟ وهذه هي:

. لإدراج صفحة إطارات نحتاج إلى ملف رئيسي يعتبر بمثابة الوعاء الذي سيضم هذه الإطارات.

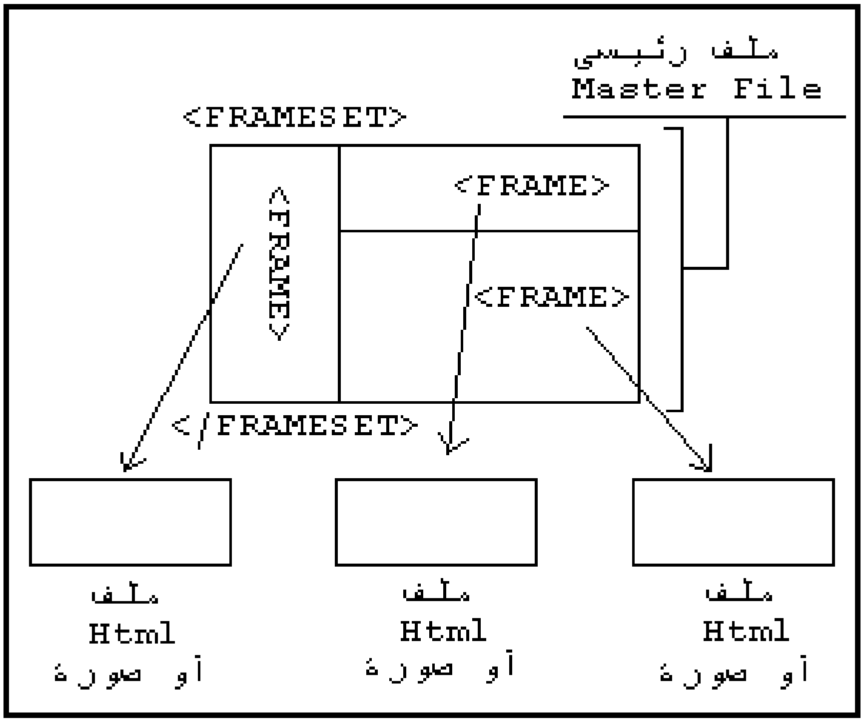
. الملف الرئيسي هو ملف HTML إعتيادي غير أننا نكتب الوسوم <FRAMESET> ...</FRAMESET> بدلاً من <BODY> ...</BODY>. وبالتالي فهو يحتوي على الخصائص التي نريدها للإطارات وتعريفاتها.

. نستخدم الخصائص COLS, ROWS لتحديد عدد الإطارات (صفوفاً كانت أو أعمدة) وأحجامها.

. الملفات الفرعية التي تظهر ضمن الإطارات هي ملفات عادية كالتي قمنا بإنشائها في الدروس السابقة أو صوراً. ولا تحتوي على أي تنسيق أو وسوم خاصة.

. نستخدم الوسم <FRAME> داخل الملف الرئيسي لمناداة الملفات الفرعية داخل الإطارات، وذلك مع الخاصية SRC. بالإضافة إلى استخدامه لتحديد باقي الخصائص .

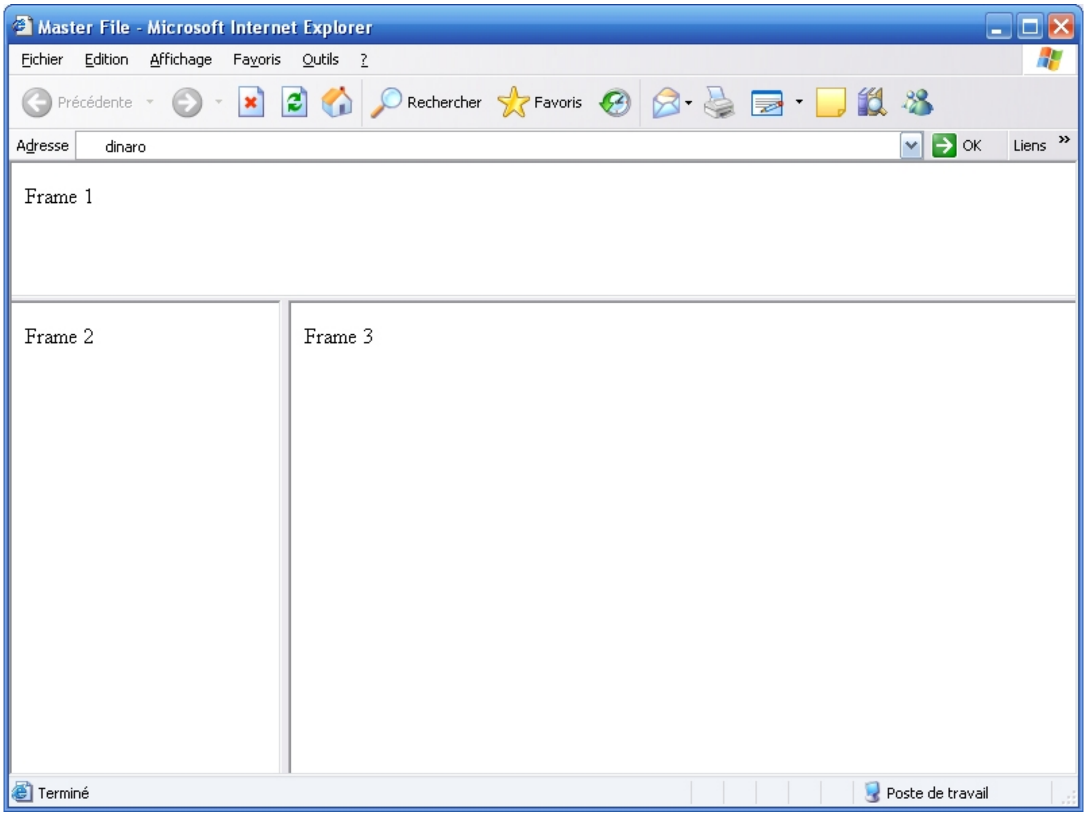
كما نستطيع تمثيل هيكلية الإطارات من خلال الشكل التالي:



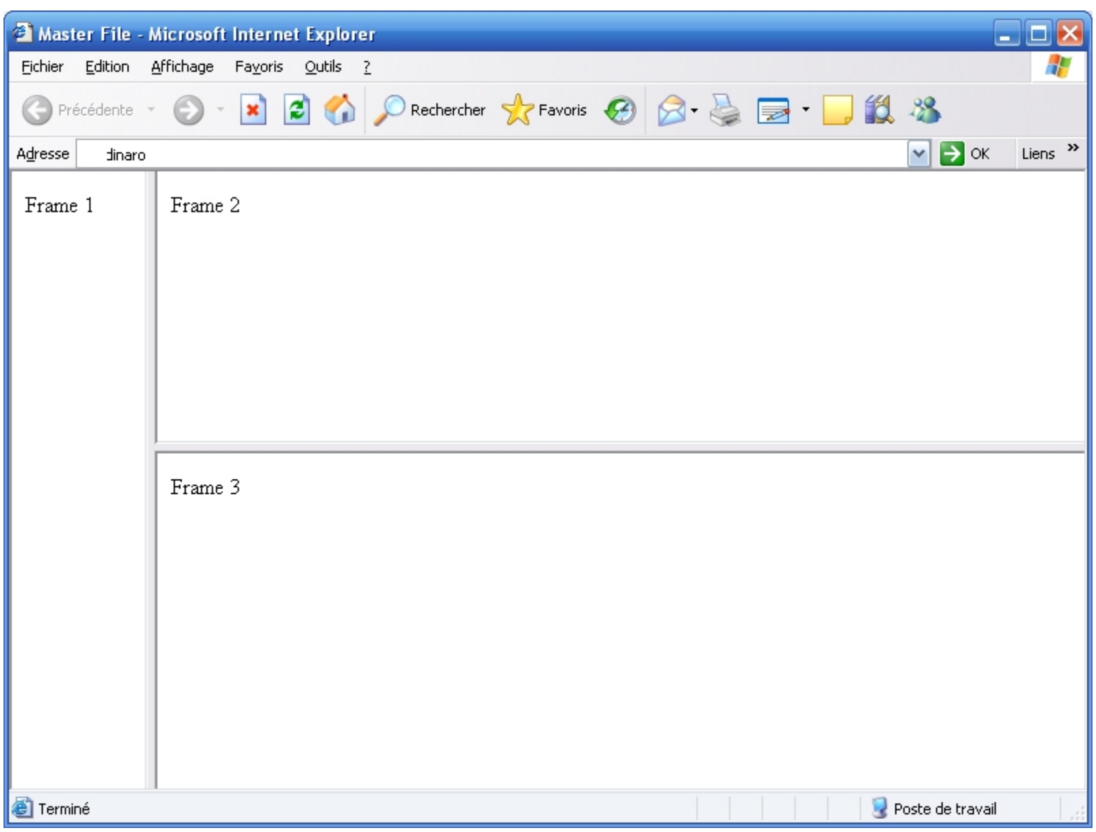
هل تأكدت من فهمك لهذه النقاط؟ لنتابع إذن ...

حتى الآن قمنا بتقسيم الصفحة إما لإطارات أفقية أو لإطارات عمودية. لكن نحتاج لمعرفة كيفية إدراج كلاهما في الصفحة. كما في الأمثلة التالية:

صفحة مكونة من صفين، الثاني منهما مقسم بدوره إلى عمودين



صفحة مكونة من عمودين، الثاني منهما مقسم بدوره إلى صفين



لنبدأ بالمثال الأول:

بما أن الصفحة تحتوي على صفين نقوم بتعريفهما أولاً حسب الإرتفاعات المرغوب بها:

```
<FRAMESET ROWS="100,*">
   <FRAME SRC="frame1.html">
   <FRAME SRC="frame2.html">
</FRAMESET>
```

لكن الصف الثاني مقسم إلى عمودين وهنا يعتبر بمفهوم لغة HTML وكأنه صفحة إطارات جديدة لذلك لا نحتاج لتعريفه كصف وبدلاً من ذلك نعاود استخدام تعريف الصفحات! أي <FRAMESET> مرة أخرى.

```
<FRAMESET ROWS="100,*">
   <FRAME SRC="frame1.html">

   <FRAMESET>
   </FRAMESET>

</FRAMESET>
```

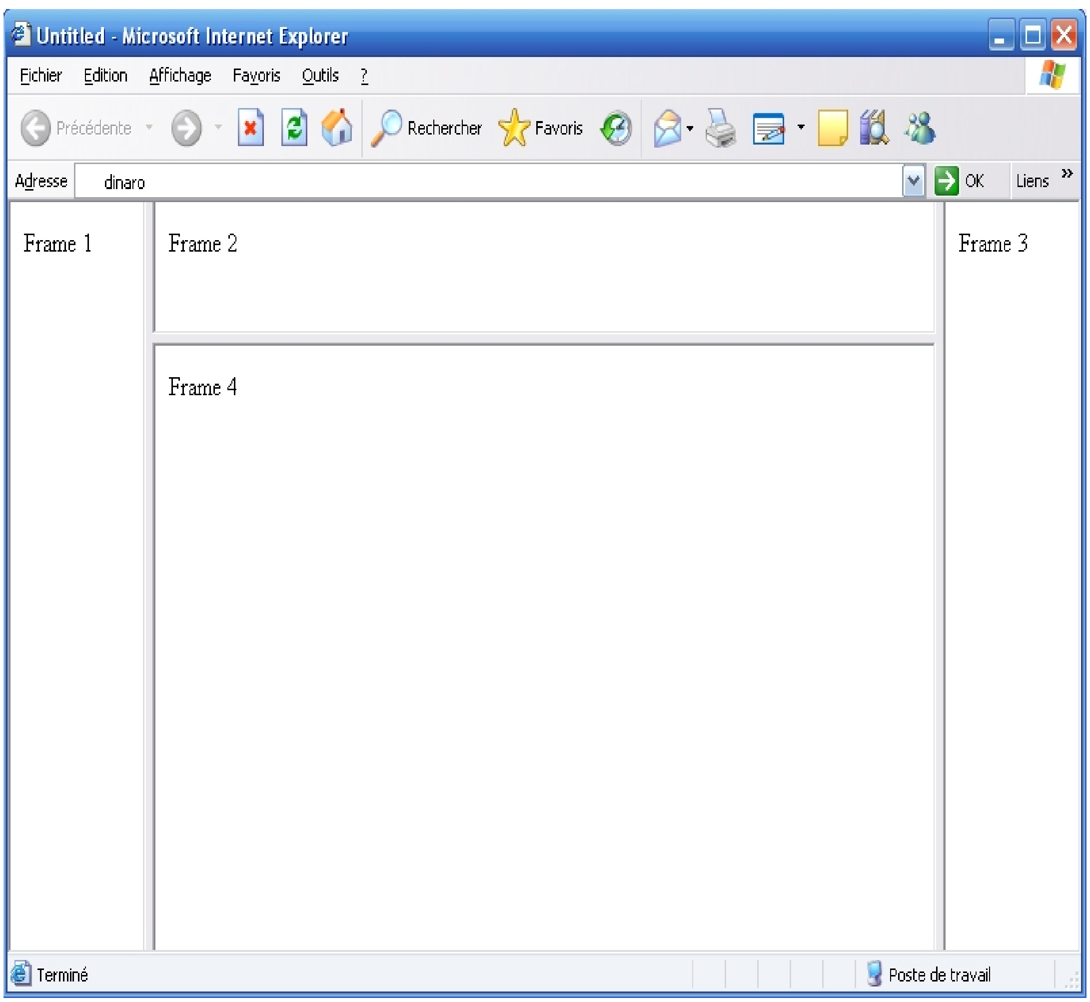
وبما أن الصف الثاني (أو لنقل الإطار الثاني) مقسم إلى عمودين، إذن بقي علينا إضافة تعريف لهذه الأعمدة. وبذلك تكون الشيفرة النهائية كالتالي:

```
<FRAMESET ROWS="100,*">
   <FRAME SRC="frame1.html">

   <FRAMESET COLS="200,*">
      <FRAME SRC="frame2.html">
      <FRAME SRC="frame3.html">
   </FRAMESET>

</FRAMESET>
```

لنقم الآن بإدراج مثال آخر وتحليله:



يوجد لدينا ثلاثة أعمدة، أليس كذلك؟ إذن لنقم بتعريف صفحة إطارات ذات ثلاثة أعمدة (طبعاً لا يوجد أهمية للأحجام المذكورة، فأنا اخترتها حسب رغبتي وتستطيع أنت اختيار الأحجام التي تريدها).

```
<FRAMESET COLS="100,*,100">
   <FRAME SRC="frame1.html">
   <FRAME SRC="frame2.html">
   <FRAME SRC="frame3.html">
</FRAMESET>
```

العمود الأوسط من هذه الصفحة مقسم إلى صفين، إذن نستبدل تعريفه بتعريف آخر لصفحة إطارات مكونة من صفين (وهذا هو التعريف بصورة مستقلة)

```
<FRAMESET ROWS="80,*">
   <FRAME SRC="frame2.html">
   <FRAME SRC="frame4.html">
</FRAMESET>
```

وبعد دمج الشيفرتين السابقتين معاً نحصل على هذه الشيفرة النهائية:

```
<FRAMESET COLS="100,*,100">
   <FRAME SRC="frame1.html">

   <FRAMESET ROWS="80,*">
      <FRAME SRC="frame2.html">
      <FRAME SRC="frame4.html">
   </FRAMESET>
```

```
   <FRAME SRC="frame3.html">
</FRAMESET>
```

وصلنا الآن إلى نـهاية هذا الدرس. أراك على خير في الدرس التالي.

ألـيس هذا ممتـعا
!!

HHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHHH

الدرس العاشر

وسوم ... من هنا وهناك!!!

أهلاً وسهلاً بك إلى الدرس العاشر من كتاب تعلم HTML.

كما ترى من العنوان فقد اخترت أن يكون هذا الدرس خارجاً قليلاً عن نطاق الدروس السابقة من حيث المحتوى. فلن تجد هنا وسوماً محددة تستطيع حصرها تحت موضوع معين، بل وسوماً عامة وخصائص إضافية لوسوم ذكرت سابقاً. وإن كان الهدف من معظمها ترتيب الصفحة والتحكم بشكل محتوياتها. ولا أخفي عليك أني قصدت تأجيل بعضها حتى هذا الدرس مع أنه كان من الممكن إدراجها ضمن دروس سابقة، وخاصة الدرس الرابع (الفقرات). لكن حجّتي في عدم إدراجها في حينه أن هذه الوسوم لن يتم استخدامها ولن تُفهم طريقة عملها إلاّ عند استخدام وسوم أخرى تمّ شرحها في وقت لاحق بعد الفقرات مثل الصور والجداول. أما البعض الآخر ففضلت عدم حصرها ضمن أي درس على اعتبار أنها وسوم عامة لا تختص بأي موضوع. على أية حال يكفينا هذه المقدمة وأترك لك حرية تصنيفها كما يحلو لك. والآن لندخل مباشرة في الموضوع.

أترى هذا الخط الذي فصلت به هذه الفقرة عن الفقرة السابقة؟ إنه يسمى بلغة HTML بالمسطرة الأفقية Horizontal Rule وتستطيع إدراجه لتقسيم صفحتك بكتابة الوسم <HR> فقط لا غير. أكتب:

```
<HR>
```

ليظهر لديك هذا الخط:

لكن هذا ليس كل شيء. لأنك تستطيع تحديد سُمك هذا الخط إذا أضفت له الخاصية SIZE وأتبعتها برقم يمثل هذا السُمك مثلاً:

<HR SIZE="5">

<HR SIZE="1">

<HR SIZE="10">

كذلك يمكنك تحديد عرض الخط باستخدام الخاصية WIDTH والتي من الممكن أن تأخذ قيمة مطلقة أو نسبية

<HR WIDTH="80%">

<HR WIDTH="400">

<HR SIZE="5" WIDTH="60%">

ومن الخصائص الأخرى لهذا الخط خاصية المحاذاه ALIGN والتي تأخذ القيم left, right ،center

<HR WIDTH="80%" ALIGN="center">

<HR WIDTH="400" ALIGN="left">

<HR SIZE="5" WIDTH="60%" ALIGN="right">

وكما تلاحظ فإن هذا الخط يكتسب مظهراً غائراً ثلاثي الأبعاد وإذا أردت خطاً عادياً غير غائر فقم بإضافة الخاصية NOSHADE

```
<HR SIZE="5" WIDTH="60%" ALIGN="center" NOSHADE>
```

أما إذا كان لون هذا الخط لا يعجبك، فما من مشكلة إذ أنك تستطيع اختيار اللون الذي يعجبك من خلال الخاصية COLOR (تعمل فقط مع MS Explorer)

```
<HR SIZE="5" WIDTH="60%" ALIGN="center"
COLOR="#FF0000" NOSHADE>
```

الوسم التالي في هذه المجموعة هو وسم الملاحظات <!-- ... --> ونستخدمه عند الحاجة لكتابة بعض الملاحظات الخاصة أو العبارات التوضيحية ضمن الملف والتي يقصد أن لا تظهر عند استعراض هذا الملف في المتصفح.

```
This is line one<BR>
<!-- This is line two --><BR>
and, this is line three<BR>
```

وهذه هي النتيجة

This is line one

and, this is line three

**من المؤكد أنك تعرف الوسم
 والذي يقوم بالتحكم في نهايات الأسطر (أي أنه ينهي السطر الحالي بحيث يظهر النص الذي يليه في سطر جديد). فهل تعلم أنه يوجد خاصية لهذا الوسم وهي CLEAR؟**

لكي تتوضح لك طبيعة عمل هذه الخاصية، قم بمراجعة ما ذكرناه في الدرس الخامس عن الصور والرسومات، واستخدام الخاصية ALIGN التي تحدد موقع هذه الصور على الصفحة. حسناً، لقد إتفقنا في حينه على أن القيمة right توجه الصورة إلى يمين الصفحة وأن النص الذي يليها يظهر ملتفاً بعدة أسطر على الجهة اليسرى. وكذلك الأمر (بصورة معكوسة) عند استخدام القيمة left. وحتى لو استخدمنا القيمة bottom أو لم نقم بإضافة الخاصية ALIGN أصلاً، فسوف نجد أن النص الذي يليها يظهر بمحاذاة الحافة السفلى للصورة.

يتلخص عمل الخاصية CLEAR في منع النص من الإلتفاف على أي من جانبي الصورة.

وهي تأخذ القيم right التي تمنع ظهور النص إلاّ عند بداية الهامش الأيمن الفارغ بعد الصورة (أي بمنتهى البساطة تمنع ظهور النص والتفافه على الجهة اليسرى للصورة... أليست هذه العبارة أسهل للفهم؟!!) وبالتالي فإن هذه القيمة تستخدم عندما تكون قيمة ALIGN للصورة هي right.

وبنفس المبدأ لكن بصورة معكوسة نستخدم القيمة left والتي في نتيجتها تمنع إلتفاف النص على الجهة اليمنى من الصورة. وذلك عندما تكون محاذاة الصورة هي left

منطق معكوس... أليس كذلك؟ على أية حال إذا كنت تكرة هذه التعقيدات وتحب الطرق المختصرة مثلي، فاستخدم القيمة all التي تمنع الإلتفاف من جميع الجوانب.

ما رأيك أن نقتبس بعض الأمثلة من الدرس الخامس لنشاهد كيف تعمل مع إضافة هذه الخاصية؟ وسوف أدرج هذه الأمثلة كما هي نصاً وحرفاً أما الإضافات فهي المميزة باللون الأحمر



والآن لنجرب استخدام القيمة left مع هذا المثال نفسه



حسناً لنستخدم القيمة left في مكانها الصحيح، أي مع المحاذاة left



وأترك لك المجال لكي تجرب القيمة all بنفسك

من القواعد الإفتراضية للمتصفحات أن الأسطر في كل فقرة تلتف وتنقسم بصورة تلقائية حسب إستبانة الشاشة وعرض نافذة المتصفح. (هذه نعرفها من الدرس الأول). لكن لنقل أننا نريد من أحد الأسطر أن لا ينقسم مهما كان مقدار الإستبانة وعرض النافذة.

حسناً، كل ما علينا فعله هو وضع هذا السطر ضمن الوسوم

<NOBR> ... </NOBR>

وهي إختصار لـِ NO BReak أي (لا إنقسام).

ها قد وصلنا إلى نهاية هذا الدرس والذي ناقشنا فيه وسوماً منوعة تتعلق في مجملها بتنسيق الصفحات. أراك على خير في الدرس التالي.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHH

الدرس الحادي عشر

النماذج [1]

أهلاً وسهلاً بك إلى الدرس الحادي عشر من كتاب تعلم HTML.

سيكون هذا الدرس الأول من درسين سنتحدث فيهما عن النماذج وكيفية تضمينها في صفحات الويب.

مع أن النماذج تعتبر من المواضيع المتقدمة (وغير السهلة) نوعاً ما في لغة HTML إلا أن معظم مواقع الويب تكاد لا تخلو من وجودها، وذلك لعدة أسباب لعل من أهمها إيجاد إمكانية للتفاعل بين الموقع وصاحبه من جهة والزوار من جهة أخرى...أحيانا قد تحتاج كمصمم لموقع ويب أن تعرف آراء زوار موقعك في مسائل معينة وقد تكتفي برسائل البريد الإلكتروني التي يرسلوها لك، لكن عندما تريد معرفة أشياء محددة بذاتها فإن النماذج هي الخيار الأفضل لك. بالإضافة إلى إمكانية تنظيم البيانات المدخلة من خلالها وسهولة وسرعة استخدامها من قبل زوار الموقع. ومن أبرز الأمثلة على النماذج في مواقع الويب هي دفاتر الزوار وصفحات البحث عن الكلمات أو العبارات داخل المواقع.

لا تكمن صعوبة التعامل مع النماذج في كونها معقدة بحد ذاتها، كلا ... فهي إحدى العناصر التي تدعمها لغة HTML وهي مجرد وسوم عادية مثلها مثل الوسوم التي تعاملنا معها في جميع الدروس السابقة. وبإمكانك إنشاء النماذج في موقعك بنفس السهولة التي تدرج فيها جدولاً أو إطاراً (هذا بالطبع إذا كنت تعتقد أن الجداول والإطارات سهلة) لكن التداخل بينها (وأعني النماذج) وبين لغات البرمجة المتقدمة في

الويب مثل JavaScript, CGI هي ما يجعلها تختلف عن سابقيها من الوسوم أو العناصر الأخرى. خاصة إذا احتجت إلى بعض المقاطع البرمجية من هذه اللغات ضمن نماذجك. أما إذا اكتفيت بالإمكانات المتواضعة التي توفرها HTML بالنسبة للنماذج. فما من مشكلة... لأنه سيكون بإمكانك التعامل معها بكل بساطة. وفي هذا الدرس لن نتطرق بالطبع إلى أي من اللغات سوى HTML.

قبل أن نبدأ، ما رأيك بزيارة صفحة (دفتر الزوار) في هذا الموقع للإطلاع على مثال للنماذج، (وربما تود أيضاً التوقيع فيه).
كم شكلاً من أشكال إدخال البيانات يوجد في هذا الدفتر؟ الحقيقة أنه يوجد ستة أشكال هي كالتالي:



وهي الأشكال الموجودة في الدفتر فقط. وأود أن ألفت نظرك إلى وجود أشكال أخرى سوف يتم التعامل معها من خلال هذا الدرس.

والآن إلى العمل يا عزيزي المتدرب

مع أن الأشكال السابقة تختلف عن بعضها البعض من حيث المبدأ والمظهر (وطريقة التعريف أيضاً) إلا أنها يجب أن تدرج جميعاً ضمن وسمين أساسيين للنماذج هما:

</FORM> ... <FORM>

وكما جرت العادة نحتاج لتحديد بعض الخصائص التي تتعلق بطبيعة هذا النموذج. ولدينا هنا ثلاث خصائص:

ACTION

تحدد العنوان الذي سيتم إرسال بيانات النموذج إليه لتتم معالجتها بالصورة المطلوبة. وعادة يكون هذا عنواناً لبريد إلكتروني Email سوف يتم إرسال بيانات النموذج إليه. أو قد يكون عنواناً لبرنامج CGI موجود على الكمبيوتر الخادم Server الذي تتواجد عليه صفحة الويب، حيث يستقبل هذه البيانات ويعالجها حسب التعليمات الموجودة فيه كأن يضيفها مثلاً إلى إحدى الصفحات (كما يحدث عادة في دفاتر الزوار) أو يتحقق من صحة بعض الحقول المدخلة ومطابقتها لمعايير معينة، أو أن يقوم بالبحث عن كلمة أو عبارة ضمن صفحات الموقع كما في نماذج البحث الموجودة في مواقع الويب.

```
<FORM ACTION="mailto:someone@domain.com"> ...
</FORM>
<FORM ACTION="name_and_address_of_CGI_script"> ...
</FORM>
```

METHOD

تحدد الطريقة التي سيتم بها التعامل مع العنوان المحدد في الخاصية السابقة ACTION. وهناك قيمتين لهذه الخاصية هما: GET التي تستخدم في حالة كون عملية المعالجة داخلية أي تتم داخل الخادم Server نفسه. ففي مثالنا السابق عندما نستخدم نموذج البحث عن كلمة في الموقع، فإن عملية المعالجة (أي البحث) تجري مباشرة في الموقع. والقيمة الثانية هي Post وتستخدم عندما تكون عملية المعالجة خارجية كأن يتم إرسال البيانات إلى عنوان بريد إلكتروني.

<FORM ACTION="mailto:someone@domain.com"
METHOD="post"> ... </FORM>
<FORM ACTION="name_and_address_of_CGI_script"
METHOD="get"> ... </FORM>

نبدأ الآن في عملية تعريف أشكال البيانات في النموذج. ونستخدم الوسم <INPUT> لتعريفها والحقيقة أن هذه الأشكال هي مجرد خصائص أو بالأحرى قيم لخصائص تابعة لهذا الوسم. كيف؟ ... لنأخذ مثالاً على ذلك لأوضح لك هذا المفهوم

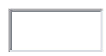
ملاحظة: إذا كنت تستخدم Sindbad x.3 فسيبدو الحقل والنص المجاور له بصورة معكوسة، وهي مشكلة ناتجة عن برنامج Netscape الذي يعمل من خلاله

:Please enter your address

حسناً، لقد استخدمت الوسم <INPUT> لتعريف هذا الشكل (هذه إتفقنا عليها مسبقاً) ومن ثم قمت بإضافة الخاصية TYPE لهذا الوسم لتحديد نوع الشكل الذي أريده وأعطيتها القيمة TEXT أي

```
<FORM ...>
<INPUT TYPE="text">
</FORM>
```

لينتج لدينا هذا الشكل:



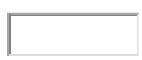
<u>فقرة معترضة</u>:
إليك جميع الأشكال (القيم) المستخدمة مع الخاصية TYPE وسوف أتركها الآن بدون تعليق لحين مناقشها لاحقاً بشكل مفصل. مع ملاحظة أن هناك شكلين آخرين ندرجهما بالوسوم <SELECT>، <TEXTAREA>

	<INPUT TYPE="text">
	<INPUT TYPE="password">
	<INPUT TYPE="hidden">
	<INPUT TYPE="radio">
	<INPUT TYPE="checkbox">
Soumettre la requête	<INPUT TYPE="submit">
Rétablir	<INPUT TYPE="reset">
	<INPUT TYPE="button">

أرجو أن أكون قد وضحت لك الآن وظيفة الخاصية TYPE وجميع القيم المستخدمة معها

ونعود الآن إلى مثالنا.. الخاصية الثانية المستخدمة مع <INPUT> هي NAME وتستخدم لتسمية حقل البيانات حيث قمت بإعطاء الإسم address لهذا الحقل في المثال. (لك كل الحرية في إعطاء الإسم الذي تريده للحقل). والحقيقة أن هذا الإسم يعرّف الحقل في داخل النموذج نفسه، بحيث يمكن استخدامه فيما بعد للحاجات البرمجية وضرورات المعالجة إن وجدت من قبل البرامج التي قد تضيفها كمصمم للموقع. وحتى عندما تريد أن يُرسل النموذج إليك بالبريد فإن حقوله تعرّف بالاسم الذي أدرجته لها من خلال هذه الخاصية. (لاحظ ما قلتُه سابقاً عن تعريف أسماء الحقول عندما تحدثنا عن الترميز والطرق التي تصل بها محتويات النموذج). وكما ترى لا يوجد (حتى الآن) ما يدل على أن هذا الحقل يختص بإدخال العنوان.

```
<FORM ...>
<INPUT TYPE="text" NAME="address">
</FORM>
```



أما العبارة Please enter your address : فهي مجرد عبارة توضيحية أضفتها ليعرف الزائر ما الذي يجب عليه كتابته وتستطيع صياغة هذه العبارة كما تريد. ففي كل الأحوال ليس لها علاقة بجوهر النموذج نفسه بعكس الخاصية NAME.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text"
```

```
NAME="address">
</FORM>
```

: Please enter your address

قد نحتاج أحياناً إلى تحديد حجم الحقل ولذلك نستخدم الخاصية SIZE مع الرقم الذي نريده كحجم للحقل، لنجرب الرقم 40

```
<FORM ...>
Please enter your address : <INPUT TYPE="text"
NAME="address" SIZE="40">
</FORM>
```

: Please enter your address

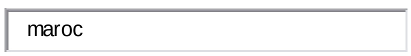
الخاصية VALUE تستخدم في الحالات التي نتوقع فيها كتابة قيمة دارجة أو متكررة من قبل معظم الزوار وللتسهيل عليهم يتم تعيينها كقيمة إفتراضية وبالطبع مع توفر إمكانية حذفها وكتابة ما يريدنه بدلاً منها.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text"
NAME="address" VALUE="maroc">
</FORM>
```

maroc : Please enter your address

لا يوجد للخاصية SIZE أي صفة تحكمية بالنسبة لحجم المدخلات التي يمكن للزائر أن يكتبها داخل الحقل. وبعبارة أخرى: صحيح أننا حددنا حجم الحقل لكن ذلك يسري فقط على مظهره على الشاشة. ولا يوجد ما يمنع الزائر من الكتابة بحيث يتجاوز النص حجم الحقل المحدد. وهنا يأتي دور الخاصية MAXLENGTH لتتحكم بالحد الأقصى للنص المدخل.

```
<FORM ...>
Please enter your address : <INPUT TYPE="text"
NAME="address" VALUE="maroc"
SIZE="40" MAXLENGTH="30">
</FORM>
```



: Please enter your address

حاول الكتابة في هذا الحقل لأكثر من 60 حرفاً وأنظر ماذا سيحدث؟

إنتهينا الآن من خصائص الوسم INPUT فما رأيك بإجمالها مرة أخرى؟ حسناً، هذه هي:

- **TYPE: لتحديد نوع (شكل) حقل البيانات.**
- **NAME: لتعيين اسم لحقل البيانات.**
- **VALUE: لتعيين قيمة إفتراضية (مبدئية) لحقل البيانات.**
- **SIZE: لتحديد حجم حقل البيانات.**
- **MAXLENGTH: لتعيين الحد الأقصى لعدد الحروف المدخلة في الحقل.**

النوع الثاني من الحقول المستخدمة في النماذج هو حقل password وهو يشبه الحقل text من حيث الخصائص تماماً غير أن مدخلاته تظهر على شكل **** مهما كانت، وهو الفرق الوحيد بينهما. وربما تكون قد استنتجت الآن أن هذا النوع من الحقول يستخدم عندما يوجد حاجة لإدخال كلمة سر من قبل الزائر في النموذج**

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE=""
SIZE="40" MAXLENGTH="30">
Please enter your passwod :
<INPUT TYPE="password" NAME="the password" VALUE=""
SIZE="40" MAXLENGTH="30">
</FORM>
```

Please enter your name :

Please enter your password :

لاحظ أنني لم أرغب في كتابة قيم إفتراضية VALUES للحقول، ولذلك تركتها فارغة وأستطيع أيضاً أن ألغيها نهائياً من الشيفرة. وأنا في هذا المثال أردت أن أوضح لك عدم أهمية كتابة قيمة إفتراضية للحقول في بعض الحالات.

نأتي الآن إلى النوع الثالث من أنواع الحقول وهو hidden أي الحقل المخفي. وكما نستنتج من اسمه فهو لن يظهر ضمن النموذج. وهذا مثال:

```
<FORM ...>
Please enter your name :
<INPUT TYPE="text" NAME="the name" VALUE=""
```

```
SIZE="40" MAXLENGTH="30">

<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">

Please enter your passwod :
<INPUT TYPE="password" NAME="the password" VALUE=""
SIZE="40" MAXLENGTH="30">
</FORM>
```

: Please enter your name
: Please enter your passwod

لاحظ هنا أن وجود هذا الحقل مثل عدمه بالنسبة لمظهر النموذج، وأن الزائر لن يتعامل معه بل وربما لن يعرف أن هناك حقلاً مخفياً. والسؤال هنا: ما الفائدة من وجود شيء مخفى لا إمكانية لاستخدامه؟ ولكي أجيب على هذا السؤال دعني أطرح لك مثالاً أو حالة قد تواجهك كمصمم صفحات ويب...

لنفرض أن لديك ثلاث صفحات تتضمن كل منها نموذجاً ما وأن هذه النماذج متشابهة. وتحتوي على نفس الحقول. وعندما تصلك البيانات كيف ستستطيع تمييز أي من هذه النماذج استخدم لإرسال البيانات؟ بإمكانك إضافة هذا الحقل (الوهمي) وإسناد أي اسم وأي قيمة له في كل نموذج.

في النموذج الأول ...

```
<INPUT TYPE="hidden" NAME="my forms" VALUE="form1">
```

في النموذج الثاني ...

<INPUT TYPE="hidden" NAME="my forms" VALUE=" form2">

في النموذج الثالث ...

<INPUT TYPE="hidden" NAME="my forms" VALUE=" form3">

وبذلك عندما تصلك البيانات المرسلة من قبل أي زائر استخدم أي من النماذج الثلاثة سيصلك أيضا حقل إضافي قمت أنت نفسك بتعبئته سلفاً عندما صممت النموذج وذلك بأحد الأشكال التالية:

my forms=form1 أو
forms=form2 my أو
my forms=form3

إذن نستطيع القول أن الحقل المخفي هو لاستخدام المصمم وليس الزائر، وأن قيمته تدخل مباشرة عند التصميم. ويستخدم بهدف تعريف قيم ما سيتم إرسالها جنباً إلى جنب ضمن بيانات النموذج التي قام الزائر بتعبئتها.

ملاحظة مهمة بالنسبة للنماذج بشكل عام. من أجل إظهار النموذج بصورة مرتبة ومنسقة والتحكم بموقع الحقول فيه فمن الأفضل دائماً وضعه داخل جدول مع جعل الجدول بلا حدود.

```
<FORM ...>
<TABLE BORDER="0">
```

```
<TR>
<TD>Please enter your name : </TD>
<TD>
<INPUT TYPE="text" NAME="the name" VALUE=""
SIZE="40" MAXLENGTH="30">
</TD>
</TR>

<TR>
<TD>Please enter your password :</TD>
<TD>
<INPUT TYPE="password" NAME="the password" VALUE=""
SIZE="40" MAXLENGTH="30">
</TD>
</TR>
</TABLE>
</FORM>
```

وكما ترى تحتاج إلى القليل من العمل الإضافي لكنك بالمقابل ستحصل على النتيجة التالية

Please enter your name :

Please enter your password :

هكذا أفضل... أليس كذلك يا أيها المصمم ؟

نكون الآن قد وصلنا إلى نهاية هذا الدرس. وأتمنى أن أكون قد نجحت في تيسير عملية فهمها والتعامل معها بالنسبة لك. ومع ذلك فنحن لم ننته بعد... فسوف نناقش ما تبقى في الدرس القادم ، ألقاك هناك.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHH

الدرس الثاني عشر

النماذج [2]

أهلاً وسهلاً بك إلى الدرس الثاني عشر من كتاب تعلم HTML.

هذا الدرس هو الثاني من درسين حول النماذج. لقد قمنا في الدرس السابق بمناقشة الوسوم الأساسية للنماذج وتعلمنا كيفية إدراج النماذج في صفحات الويب. كما قمنا بمناقشة بعض أشكال إدخال البيانات في النموذج وهي Text, Password, Hidden هل تذكر كيف نقوم بتعريفها؟ راجع الدرس السابق إن أردت المزيد من التوضيح، وإلا هيا بنا نكمل ولندخل في الموضوع مباشرة.

سوف نتابع الآن مع مجموعة من الأشكال الخاصة بالاختيار من متعدد وهي بالمناسبة ثلاثة أنواع: Radio, Checkbox وقائمة الإختيار

نبدأ مع الشكل المسمى RADIO. ومن مسوغات استخدام هذا الشكل أن السؤال المطروح ينبغي أن يكون له إجابة واحدة فقط، أو بعبارة أخرى على الزائر أن يختار إجابة واحدة فقط.

وكمثال، لنفرض أنني أريد أن أسأل الزائر عن المتصفح الذي يستخدمه (كما هو موجود في دفتر الزوار على شكل قائمة إختيار) لكن بدلا من أن يكون على شكل قائمة إختيار

أريده أن يكون على شكل RADIO وذلك بالشكل التالي:(أود أن أذكرك أن عناصر النموذج تظهر بشكل معكوس إذا كنت تستخدم Sindbad 3.0)

- Sindbad 3.0
- Sindbad 4.0
- Ms Explorer 3.0
- Ms Explorer 4.0

فكيف ننشيء مثل هذه القائمة؟ ... حسناً، لنبدأ من الصفر ونعرّف نموذجاً

```
<FORM>

</FORM>
```

ثم لنقم بتعريف هذا الشكل، هل تذكر الوسم الخاص بذلك؟ إنه <INPUT>

```
<FORM>
<INPUT TYPE="radio">
</FORM>
```

لكن بما أن هناك أربعة مدخلات، إذن نحتاج إلى أربعة وسوم

```
<FORM>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
<INPUT TYPE="radio"> <BR>
```

```
<INPUT TYPE="radio"> <BR>
</FORM>
```



نحتاج الآن إلى تسمية هذه المدخلات، أي أننا سنستخدم الخاصية **NAME** معها. أما الاسم المعطى بحد ذاته فمن الأفضل أن يكون مرتبطاً نوعاً ما بموضوع السؤال، ليس لأن هذا ضروري للنموذج بل هو ضروري لك كشخص سيقوم باستقبال البيانات المرسلة من خلال النموذج، وبالتالي من الأفضل أن يوجد عنوان معبّر للبيانات بغرض التمييز. وبما أننا هنا نتحدث عن المتصفحات فليكن هذا الاسم هو **browser**

```
<FORM>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
<INPUT TYPE="radio" NAME="browser"> <BR>
</FORM>
```



وكما تلاحظ من النتيجة أن هذه التسمية هي ضمنية فقط ولا تؤثر على شكل النموذج (راجع الدرس السابق) لكن أي إختيار سيقوم به الزائر من هذه الأربعة خيارات سوف يصلك تحت الاسم **browser**.

الخطوة التالية هي إعطاء قيمة لكل مدخلة في هذه القائمة وذلك حسب ما نراه مناسباً، إذن حان الوقت لاستخدام الخاصية VALUE:

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3">
<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4">
<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3">
<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4">
<BR>
</FORM>
```

وهنا أيضا نلاحظ أن لا تغيّر في شكل النموذج ظاهرياً مع إضافة هذه الخاصية. لكن مع ذلك فقد قمنا حتى الآن بتسمية الحقول وإعطاء كل حقل قيمة محددة. وفعلياً لقد إنتهينا من هذا النموذج. لكن بالطبع نحن لا نتوقع أن يكون الزائر عالماً بالغيب لكي يخمن أي من هذه الحقول تختص بكل قيمة. لذلك بقى علينا تعريف كل حقل باسم صريح يوضح محتواه.

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3">
Sindbad 3.0 <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4">
Sindbad 4.0<BR>
```

```
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3">
MS Explorer 3.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4">
MS Explorer 4.0<BR>
</FORM>
```

- Sindbad 3.0
- Sindbad 4.0
- MS Explorer 3.0
- MS Explorer 4.0

وهناك خاصية تتعلق بهذا النوع من الحقول، وهي أنك إذا أردت أن يظهر أحدها وقد تم اختياره بشكل تلقائي فعليك بإضافة الخاصية CHECKED إليه، مع ترك كل الحرية للزائر في اختيار ما يريده فيما بعد.

```
<FORM>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind3">
Sindbad 3.0 <BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Sind4"
CHECKED>
Sindbad 4.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie3">
MS Explorer 3.0<BR>
<INPUT TYPE="radio" NAME="browser" VALUE="Msie4">
MS Explorer 4.0<BR>
</FORM>
```

- ○ Sindbad 3.0
- ◉ Sindbad 4.0
- ○ MS Explorer 3.0
- ○ MS Explorer 4.0

وأخيراً... أود أن أوضح لك الصورة التي يصلك بها النموذج عند اختيار أحد حقوله (ولنفترض أنه الخيار الثالث). وهي بالشكل التالي:

browser=Msie3

نأتي الآن إلى الشكل الثاني من أشكال الإختيار من متعدد والذي يدعى CHECKBOX. ظاهرياً لا يختلف هذا الشكل عن الشكل الذي سبقه، لكن عملياً هناك اختلافات جذرية من حيث المفهوم والتعريف. وأنا أفضّل أن نبقى على استخدامنا للمثال السابق حتى يسهل علينا تمييز الفروق.

- ☐ Sindbad 3.0
- ☐ Sindbad 4.0
- ☐ MS Explorer 3.0
- ☐ MS Explorer 4.0

قبل أن نستمر قم بالنقر على أكثر من حقل في القائمة السابقة وأنظر ماذا سيحدث؟ إن باستطاعتك اختيار أكثر من حقل في نفس الوقت! وهذا هو الفرق الأول بين CHECKBOX و RADIO ففي RADIO يمكن اختيار حقل واحد فقط ليس أكثر.

```
<FORM>
<INPUT TYPE="checkbox" NAME="Sind3" VALUE="Yes">
Sindbad 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Sind4" VALUE="Yes">
Sindbad 4.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie3" VALUE="Yes">
MS Explorer 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie4" VALUE="Yes">
MS Explorer 4.0 <BR>
</FORM>
```

ماذا تلاحظ؟ أولاً لقد أسندنا القيمة checkbox للخاصية TYPE. ثم أعطينا لكل حقل في القائمة إسماً مميزاً في الخاصية NAME يختلف عن باقي الحقول. أما الخاصية VALUE فأعطيناها قيمة موحدة لجميع الحقول. وبالطبع قمنا في النهاية بكتابة الأسماء التعريفية لكل حقل.

في RADIO نستطيع اختيار حقل واحد فقط أما في CHECKBOX فنختار أكثر من حقل، لذلك يستخدم عادة في الحالات التي يحتمل أن نحصل فيها على عدة أجوبة لنفس السؤال.

في RADIO تكون أسماء الحقول موحدة والقيم مختلفة، أما في CHECKBOX فتكون الأسماء مختلفة والقيم موحدة

كيف ستصل البيانات؟ حسناً لنفرض أنه تم اختيار الحقلين الثاني والرابع فسوف تصلك النتيجة بالشكل التالي:

```
Sind4=Yes
Msie4=Yes
```

كما نستطيع أيضاً تعليم بعض الحقول بصورة تلقائية كما فعلنا مع RADIO باستخدام نفس الخاصية CHECKED

```
<FORM>
<INPUT TYPE="checkbox" NAME="Sind3" VALUE="Yes"
CHECKED>
Sindbad 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Sind4" VALUE="Yes">
Sindbad 4.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie3" VALUE="Yes"
CHECKED>
MS Explorer 3.0 <BR>
<INPUT TYPE="checkbox" NAME="Msie4" VALUE="Yes">
MS Explorer 4.0 <BR>
</FORM>
```

- [x] Sindbad 3.0
- [ ] Sindbad 4.0
- [x] MS Explorer 3.0
- [ ] MS Explorer 4.0

النوع الثالث من أشكال الإختيار من متعدد هو قوائم الاختيار، وهذا النوع سوف يقودنا إلى وسوم جديدة من وسوم التعريف والتي ستستخدم بدلاً من <INPUT> وهي

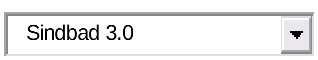
```
<SELECT>
   <OPTION>
   <OPTION>
```

```
   <OPTION>
   .....
   .....
</SELECT>
```

بحيث أن <SELECT> ... </SELECT> تحددان بداية ونهاية القائمة، والوسم <OPTION> الذي يوضع دائما بينهما يستخدم لتحديد كل عنصر من عناصر القائمة. لنعد إلى مثالنا السابق لنرى كيف يمكن وضع الخيارات في قائمة

```
<FORM>
<SELECT>
   <OPTION> Sindbad 3.0
   <OPTION> Sindbad 4.0
   <OPTION> MS Explorer 3.0
   <OPTION> MS Explorer 4.0
</SELECT>
</FORM>
```

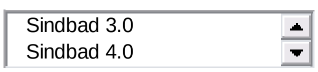
وبذلك تكون النتيجة هي:



وكالمعتاد لا يخلو الأمر من وجود خصائص تحدد طريقة عمل هذه الوسوم. وهناك خصائص مشتركة عرفناها في الأشكال السابقة سيتم استخدامها هنا أيضاً كما يوجد خصائص جديدة تتعلق فقط بهذا الشكل من حقول البيانات. فبالنسبة لِـ <SELECT> يوجد الخاصية NAME وهي كما تعلم تحدد اسم القائمة. كما توجد الخاصية SIZE التي تحدد حجم (أو بالأحرى) ارتفاع القائمة، وبالتالي عدد البيانات الظاهرة فيها. وهي تأخذ أي قيمة عددية صحيحة.

```
<FORM>
<SELECT NAME="browser" SIZE="2">
   <OPTION> Sindbad 3.0
   <OPTION> Sindbad 4.0
   <OPTION> MS Explorer 3.0
   <OPTION> MS Explorer 4.0
</SELECT>
</FORM>
```

وبذلك تكون النتيجة هي:

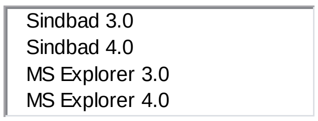


وطالما بالإمكان عرض القائمة بأي ارتفاع نريد، وقد يصل إلى حد عرض جميع بيانات القائمة معاً، فإن هناك إمكانية أيضاً لجعل اختيار البيانات من هذه القائمة متعدداً وليس فقط قيمة واحدة، كيف؟؟ بإضافة الخاصية MULTIPLE. لنقم الآن بعرض جميع القيم (لدينا أربعة قيم، إذن القيمة المكتوبة مع SIZE يجب أن تكون 4)، ومن ثم لنتح المجال أمام الزائر لاختيار أكثر من قيمة واحدة في القائمة.

```
<FORM>
<SELECT NAME="browser" SIZE="4" MULTIPLE>
   <OPTION> Sindbad 3.0
   <OPTION> Sindbad 4.0
   <OPTION> MS Explorer 3.0
   <OPTION> MS Explorer 4.0
```

```
</SELECT>
</FORM>
```

وبذلك تكون النتيجة هي:



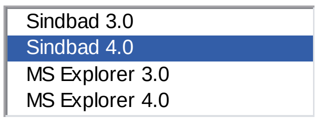
لاحظ أنه لأداء عدة اختيارات يجب أن تقوم بالضغط على المفتاح ctrl بصورة متواصلة أثناء عملية الإختيار.

أما الخصائص المستخدمة مع الوسم <OPTION> فهي VALUE والتي استخدمناها من قبل وسنستخدمها الآن لإعطاء قيمة لكل حقل بيانات في القائمة. وكذلك الخاصية SELECTED والتي نكتبها مع أي <OPTION> نريد أن يظهر وقد تم اختياره بصورة تلقائية.

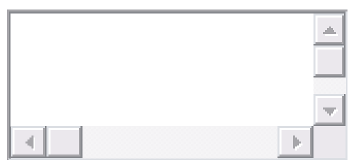
```
<FORM>
<SELECT NAME="browser" SIZE="4" MULTIPLE>
   <OPTION VALUE="Sindbad 3.0"> Sindbad 3.0
   <OPTION VALUE="Sindbad 4.0" SELECTED> Sindbad
    4.0
   <OPTION VALUE="MS Explorer 3.0"> MS Explorer 3.0
```

```
   <OPTION VALUE="MS Explorer 4.0"> MS Explorer 4.0
</SELECT>
</FORM>
```

وبذلك تكون النتيجة هي:



الشكل التالي من أشكال حقول البيانات يدعى TEXTAREA

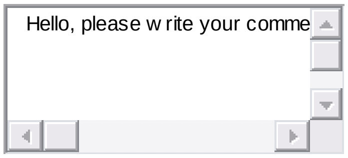


وهو المستخدم عادة لكتابة التعليقات الحرة في النموذج ويتم إدراجه بكتابة الوسوم

<TEXTAREA> ... </TEXTAREA>

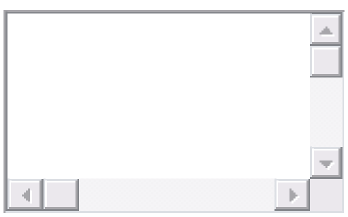
هل تستطيع تخمين الخصائص المستخدمة معه؟ بالطبع لا بد من وجود الخاصية NAME لإعطاءه اسم التعريف. لكن لا وجود للخاصية VALUE ، وبالمقابل فإن أي نص يكتب بين الوسمين سيتم عرضه داخل الحقل بصورة تلقائية

```
<TEXTAREA NAME="comments">
Hello, please write your comments here :-)
</TEXTAREA>
```



كما توجد خصائص لتحديد مساحة هذا الحقل عرضاً وارتفاعاً، وهي COLS التي تحدد العرض و ROWS التي تحدد الإرتفاع

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6">
</TEXTAREA>
```



أما الخاصية الأخيرة هنا فهي WRAP التي تحدد طريقة إلتفاف النص المكتوب داخل الحقل (لا تعمل هذه الخاصية مع MS Explorer 3.0) وهناك ثلاثة قيم تأخذها وهي على النحو التالي:

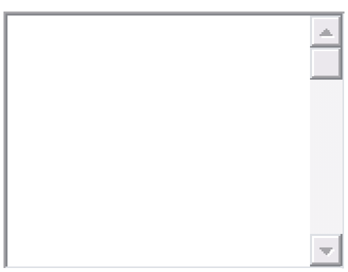
virtual : التي تعني أن النص سيلتف على عدة أسطر عند كتابته ولكنه سيصلك عند إرساله على شكل سطر واحد متتابع (حاول الكتابة داخل الحقول وأنظر كيفية تأثير هذه الخاصية على كل منها)

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6"
WRAP="virtual">
</TEXTAREA>
```



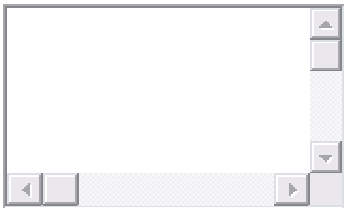
physical : تعني أن النص سيلتف على عدة أسطر وسيصلك أيضاً على هذا النحو عند إرساله

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6"
WRAP="physical">
</TEXTAREA>
```



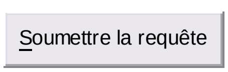
off : تعني أن النص لن يلتف بصورة تلقائية على عدة أسطر لكنه على أية حال سيصلك بنفس الشكل الذي تم إدخاله به

```
<TEXTAREA NAME="comments" COLS="30" ROWS="6"
WRAP="off">
</TEXTAREA>
```



حسناً، بافتراض أننا إنتهينا من كتابة الشيفرة الخاصة بالنموذج وننتظر من أي زائر للموقع أن يملأه، والسؤال هو كيف يمكن له أن يرسله فعلياً؟ نعود الآن إلى الوسم <INPUT> وهذه المرة مع النوع submit والتي ستقوم تلقائياً بإنشاء زر سيقوم عند النقر عليه بإرسال البيانات التي تم ملؤها في النموذج.

```
<INPUT TYPE="submit">
```



لاحظ أن Submit أو (Submit Query في Netscape) ظاهرة على الزر وهي العبارة الإفتراضية، فإذا أردت تغييرها فعليك باستخدام الخاصية VALUE لهذا الغرض

```
INPUT TYPE="submit" VALUE="Press here to send the
form">
```

Press here to send the form

في حالة كان زوار موقعك من النوعية المترددة من الناس والذين قد يغيرون آرائهم في آخر لحظة، يمكنك أن تتيح لهم إمكانية مسح ما كتبوه في النموذج وإلغاء الأمر، وذلك باستخدام reset كنوع TYPE للوسم <INPUT> بنفس طريقة التعريف والخصائص المستخدمة مع submit.

<INPUT TYPE="reset" VALUE="Forget about it">

Forget about it

الشكل الأخير من أشكال البيانات في النماذج والمدرج مع الوسم <INPUT> هو button والذي يقوم بإنشاء زر ضمن النموذج، وهو مرتبط بالنماذج التي تحتوي على نصوص برمجية (أو برامج مكتملة) من لغات متقدمة مثل JavaScript كونه يستخدم لتشغيل هذه البرامج وإطلاقها. وطبعاً هناك طرق معينة لربطها مع البرامج وليس هنا المجال لطرحها. لكن مبدئياً أقول إن طريقة الإدراج والتعريف هي ذاتها المستخدمة مع reset, submit.

<INPUT TYPE="button" VALUE="This is a sample button">

وأخيراً ... وصلنا الآن إلى نهاية حديثنا عن النماذج. فما رأيك؟ هل هو من المواضيع السهلة أم الصعبة؟ لا شيء سهل في بدايته. لذلك من الأفضل لك أن تحاول دائماً التدرب أولاً بأول على الوسوم المشروحة، بل والعودة إلى الدروس السابقة إذا اقتضى الأمر وخاصة إذا تداخلت بعضها مع الدروس الأحدث.

إلى اللقاء ... ومع تمنياتي لك بنماذج موفقة تخلو من التعقيد.

أليس هذا ممتعا

!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
!!!!!!!!!!!!!!!!!!!!!!!!!!!!

HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHHHHHHHHHH
HHHHHH

الدرس الثالث عشر

الوسوم الخاصة ب MS Internet Explorer

أهلاً وسهلاً بك إلى الدرس الثالث عشر من كتاب تعلم HTML.

والذي سنناقش فيه واحداً من المواضيع المهمة.. والشائكة.. والمزعجة نوعاً ما في هذه اللغة، وفي مجال تصميم صفحات الويب عموماً. وهو موضوع الوسوم الخاصة... فما هي هذه الوسوم؟

ملاحظة مهمة: تحتوي هذه الصفحة على وسوم خاصة بالمتصفح MS Internet Explorer فإذا كان متصفحك خلاف ذلك فلن تستطيع مشاهدة الصفحة بالصورة المناسبة

. أصوات موسيقية <BGSOUND>

. لافتات <MARQUEE>

<BGSOUND>

تستطيع تعيين أي ملف صوتي يحمل الإمتداد mid أو wav أو au بحيث يعمل تلقائياً عندما يتم تحميل الصفحة في المتصفح، وذلك باستخدام الوسم <BGSOUND> (أي Sound BackGround) فعلى سبيل المثال، قمت بإدراج المقطع المسمى lkl.mid بالطريقة التالية

```
<BGSOUND SRC="lkl.mid">
```

إذن هناك الخاصية SRC والتي تستخدم كما ترى لتحديد اسم الملف المدرج. كما يوجد أيضاً الخاصية LOOP التي تحدد عدد مرات تكرار عزف المقطوعة أي أنها تأخذ أرقاماً صحيحة عادة. أما لو قمت بإعطاءها القيمة 1- أو القيمة infinite فهذا سوف يؤدي إلى تكرار عزف المقطوعة إلى ما لا نهاية (لا تخف، فهذا لا يعني أن العزف سيستمر بعد أن تقوم بإطفاء جهازك) فقط سوف تسمعها طالما أنك موجود في الصفحة وسيتوقف بمجرد انتقالك لصفحة أخرى أو خروجك من البرنامج.

```
<BGSOUND SRC="lkl.mid" LOOP="infinite">
<BGSOUND SRC="lkl.mid" LOOP="3">
```

أما في حالة أردت إدراج وصلات تشعبية لملفات صوتية في موقعك بحيث يمكن للزائر أن يحمّلها أو أن ينقر عليها إذا أراد سماعها، فذلك يتم بالطريقة الإعتيادية لإدراج الوصلات والتي تعلمناها في الدرس السادس أي كما يلي:

```
<A HREF="lkl.mid"> ... </A>
```

</MARQUEE> ... <MARQUEE>

نص يتحرك من اليسار إلى اليمين

ما رأيك بهذا النص المتسكع أمامك على الشاشة؟ جميل ..أليس كذلك؟ هذه اللافتة (إن جاز التعبير) هي إحدى المؤثرات الخاصة التي يمكن إحداثها في MS Explorer. بواسطة هذه الوسوم. وبكل بساطة هذه هي الشيفرة الخاصة بذلك

```
<MARQUEE>
HTML
</MARQUEE>
```

بالطبع لا يمكن أن تمر الأمور هنا ببساطة فلا بد من وجود خصائص لهذه الوسوم، والحقيقة أن هناك إحدى عشرة خاصية نستخدمها (عدد ليس بالقليل بالنسبة لوسوم خاصة). لكنها على أية حال ليست صعبة التطبيق بل إننا تعاملنا بشكل أو بآخر معها مسبقاً وخاصة مع الوسم <IMG> وهذه هي الخصائص أسردها لك مع الأمثلة:

BGCOLOR: لتحديد لون الخلفية

```
<MARQUEE BGCOLOR="#FF0000">
HTML
</MARQUEE>
```

HEIGHT: لتحديد إرتفاع اللافتة، وهي إما أن تأخذ قيمة ثابتة أو قيمة نسبية حسب إرتفاع نافذة المتصفح.

```
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80">
HTML
</MARQUEE>
```

WIDTH: لتحديد عرض اللافتة، وهنا أيضاً إما أن تأخذ قيمة ثابتة أو نسبية

```
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80"
WIDTH="40%">
HTML
</MARQUEE>
```

ALIGN: لتحديد المحاذاة العمودية للنص الذي قد يتواجد على جانبي اللافتة، وهي تأخذ القيم top, middle و bottom وهي الإفتراضية أي التي تطبق تلقائياً عند عدم إدراج هذه الخاصية

```
Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80"
WIDTH="40%">
HTML
</MARQUEE>
Have a good time.
```

Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" ALIGN="middle">
HTML
</MARQUEE>
Have a good time. Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" ALIGN="middle">
H.P in arabic
</MARQUEE>
Have a good time.

Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" ALIGN="top">
HTML
</MARQUEE>
Have a good time.

HSPACE: لتحديد المسافة الفارغة التي تفصل بين اللافتة وبين النص الموجود على جانبيها

Welcome to
<MARQUEE BGCOLOR="#FF0000" HEIGHT="80" WIDTH="40%" HSPACE="30">
HTML
</MARQUEE>
Have a good time

VSPACE: تعمل مثل الخاصية السابقة لكنها تحدد المسافة الفارغة عمودياً من الأعلى والأسفل

حتى الآن كانت الخصائص التي ناقشناها مجرد خصائص مظهرية تتعلق بشكل ومظهر اللافتة دون تغيير طريقة عملها. والآن نأتي إلى الخصائص الفنية:

BEHAVIOR تحدد سلوك النص الموجود في اللافتة وطريقة عمله وهناك ثلاثة قيم (أو بالأحرى) ثلاثة أساليب لحركة النص وهي التالية:

- **scroll يتحرك بنفس الإتجاه من جانب إلى آخر وبصورة مستمرة وهي القيمة الإفتراضية، لذلك لا يهم إن كتبت أم لا.**

- **slide يتحرك النص مرة واحدة من جانب إلى الجانب الآخر ويتوقف عنده.**

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%"
BEHAVIOR="slide">
HTML
</MARQUEE>
```

من المحتمل أن يكون النص في المثال متوقفاً بعد أن يكون قد أكمل تحركه، لذلك قم بالنقر على زر Refresh الموجود في متصفحك لإعادة تحميل الصفحة وتحريك النص مرة أخرى

- **alternate يتأرجح النص جيئة وذهاباً من جانب إلى آخر.**

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%"
BEHAVIOR="alternate">
HTML
</MARQUEE>
```

DIRECTION تحدد إتجاه سير النص وذلك من خلال القيم left التي تحركه إلى اليسار (وهي الإفتراضية) و right التي تحركه إلى اليمين.

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%"
DIRECTION="right">
HTML
</MARQUEE>
```

LOOP تحدد عدد المرات التي سيتحرك فيها النص داخل اللافتة. فإذا أردت أن تستمر الحركة إلى ما لا نهاية فضع القيمة 1- أو infinite وبخلاف ذلك ضع عدد المرات التي تريدها. والشيفرة التالية تحدد عدد مرات الحركة بثلاث. (إذا وجدت النص ساكناً أو لم تجد نصاً على الإطلاق فهذا يعني أنه قد استنفذ حركاته الثلاث لذلك تحتاج للنقر على Refresh لإعادة تشغيله)

```
<MARQUEE BGCOLOR="#FF0000" WIDTH="80%"
DIRECTION="right" LOOP="3">
HTML
</MARQUEE>
```

بقي لدينا الآن خاصيتين تقومان بتحديد سرعة تحرك النص في اللافتة

والآن إليك الخصائص:

SCROLLAMOUNT: هذه الخاصية تحدد المسافة بين كل لقطة وأخرى للنص وهي تأخذ قيماً عددية صحيحة تحدد المسافة المقطوعة بالبيكسل (أعرف أنها صعبة قليلاً لكن لا بأس سأوضحها بالأمثلة).

```
<MARQUEE SCROLLAMOUNT="1"> HTML
</MARQUEE>
```

```
<MARQUEE SCROLLAMOUNT="50"> HTML
</MARQUEE>
```

```
<MARQUEE SCROLLAMOUNT="100"> HTML
</MARQUEE>
```

<MARQUEE SCROLLAMOUNT="200"> HTML
</MARQUEE>

في الحالة الأولى فإن المسافة التي تقطعها كلمة HTML في كل حركة لها هي 1 بيكسل. أما في المثال الثاني فالمسافة هي 50 بيكسل أي أن الكلمة تقفز 50 بيكسل في كل خطوة (أو لقطة) بما يساوي 50 ضعفاً عن المثال السابق وهذا ما يعطي الإنطباع بالسرعة، وهكذا الأمر للمثالين اللاحقين وهما 100 و 200 بيكسل على التوالي.

SCROLLDELAY: لكي نحدد الزمن الذي يستغرقه النص في القفز بين كل خطوة وأخرى نستخدم هذه الخاصية، والقيمة المعطاه مع هذه الخاصية تمثل الزمن بالميلي ثانية (0.001 من الثانية)

سوف أستخدم نفس الأمثلة السابقة مع هذه الخاصية لنقارن بين النتائج في الحالتين:

<MARQUEE SCROLLAMOUNT="1"
SCROLLDELAY="500"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="50"
SCROLLDELAY="500"> HTML </MARQUEE>

<MARQUEE SCROLLAMOUNT="100"
SCROLLDELAY="500"> HTML </MARQUEE>

```
<MARQUEE SCROLLAMOUNT="200"
SCROLLDELAY="500"> HTML </MARQUEE>
```

لقد أسندت القيمة 500 لهذه الخاصية في جميع الأمثلة وهي تعني أن هناك فترة نصف ثانية (بالتمام والكمال) تفصل بين كل خطوة وأخرى للكلمة. وأعتقد أن ما تراه الآن يوضح لك مبدأ عمل الخاصية السابقة أيضاً. وبالمناسبة، قد تضطر للإنتظار دهراً كاملاً لكي ترى النص في المثال الأول.

ما رأيك الآن لو نلغي الخاصية SCROLLAMOUNT لنرى كيف تعمل SCROLLDELAY لوحدها

```
<MARQUEE SCROLLDELAY="500"> HTML </MARQUEE>
```

نكون الآن قد وصلنا إلى نهاية هذا الدرس وهذا الكتاب، مع أصدق تمنياتي بأن يكون موقعك دائما منيرااااااااااااا بعون الله.........

النهاية

المراجع التي إستفدت منها هذا الكتاب :

موقع :

www.khayma.com\HPINARABIC

تم بحمد الله

نسأل الله تبارك وتعالى أن يكون هذا العمل طاهرا إنه ولي ذلك والقادر عليه.

وصلي اللهم وسلم وبارك على نبينا محمد وعلى آله وصحبه أجمعين.

وآخر دعوانا أن الحمد لله رب العالمين.

والسلام عليكم ورحمه الله وبركاته.